IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 16, 2019, accepted March 28, 2019, date of publication April 3, 2019, date of current version April 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909011

A Trusted-ID Referenced Key Scheme for Securing SCADA Communication in Iron and Steel Plants

JUNLEI QIAN[1,2], CHANGCHUN HUA[1], XINPING GUAN[3], (Fellow, IEEE), TIEFENG XIN[4], AND LIMIN ZHANG[5]

[1]Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
[2]College of Electrical Engineering, North China University of Science and Technology, Tangshan 063000, China
[3]School of Electronics, Information and Electric Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[4]China Electric Power Hwaray Technology Co., Ltd., Beijing 100094, China
[5]Department of Mathematics and Computer Science, Hengshui University, Hengshui 053000, China

Corresponding author: Junlei Qian (catherineqjl@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703149.

ABSTRACT Supervisory control and data acquisition (SCADA) is a widely implemented structure to achieve remote measurement and control in many iron and steel plants. In traditional consideration, more attention on physical network separation methods is paid to isolate the SCADA system from management network to keep SCADA in a considered ''safe'' state. In addition, lots of security solution providers are focusing on the network side security assurance without involving the SCADA communication level protection. This paper investigates a new trusted-ID referenced key scheme for securing SCADA communications efficiently. The advanced encryption standard algorithm is used in the data transmission for its fast calculating speed, and the elliptic curve cryptography digital signature algorithm is used to confirm the data package that is from the right ID which can avoid the measured values and the control instructions to be maliciously modified by attacker. This solution for securing SCADA communication provides an efficient way to protect the data and protocol between the controllers and the remote terminal units (RTUs), and offers an authentication for the communication, which can avoid Man-In-The-Middle attack. Random numbers are used as a session key that can avoid the replay attack. cipher-block chaining mode message authentication code calculation is used to meet the data integrity requirement. Gong Needham Yahalom logic is used to prove the security of this solution, and an example is given to verify its validity.

INDEX TERMS SCADA, communication security, key scheme, trusted-ID referenced scheme, ECC, digital signature.

I. INTRODUCTION

Iron and steel plants are typical process manufacturing industry based on the synergy of material flow, energy flow and information flow. The data transmission and exchage play an important role in the whole process. SCADA (Supervisory Control And Data Acquisition) system is widely used in iron and steel plants to deal with this massive amount of data in very short period of time. The data include the value measured from sensors such as the measurement of pressure, flow, temperature of gas, water and the control instructions the controller send to the actuators e.g. the valves etc. that are very important to the control system. Once the data is tampered by adversaries, the results would be severe, e.g the control process can be ruined by maliciously manipulated instructions which will result in massive physical damage [1] or the abnormal value will change the production plan. So the data security in SCADA system is the vornerstone for the process control or even the entire plant.

The security of SCADA systems are lagging behind the development of internet both in the intrusion detection for the networks and in the communication secure protection. Physical isolate the SCADA system from management network is the usual measure to keep the SCADA systems in a considered 'safe'. But this confidence of physical isolation can be collapsed by unknown U disk or disgruntled employee. In 2015 NIST (National Institute of Standards and Technology) advised that SCADA systems should be designed to include encryption and authentication between devices in order to make it very difficult to reverse engineer protocols

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Yang.

and forge packets on control system networks to avoid MITM (Man-In-The-Middle) Attacks [2].

The vulnerabilities in the protocols in SCADA systems as K. C. Mahapatra analyzed in [3] that communication protection in SCADA systems has been dramatically ignored, and this will cause fatal damages and losses [4]. PLCs are the main controllers in the SCADA systems in iron and steel plants, and MITM Attack and Replay Attack are listed to be the common threats to PLCs not dealing with encrypted packets in [5]. Other main threats in SCADA systems are listed in [6]: APTs (Advanced Persistent Threats), Lack of Data Integrity and DoS (Denial of Service) Attacks.

More and more security solutions emerged to prevent the SCADA system from being attacked. Apart from anti-virus and intrusion detection systems for network solutions [7], [8], some encryption proposals to secure the SCADA communication with different key exchange schemes are discussed.

In [9], [10], Dawson *et al.* proposed master key preloaded mode using symmetric algorithm, and a same master key should be loaded first into each SCADA communication entities so as to assure generating session key from this same master key. Symmetric encryption is used in Kang *et al.*'s solution [11], but it raised a contradiction between the frequency to change the session key and the net traffic which will probably cause the master key be exposed or time delay even a communication failure. Kang proposed QoS (Quality of Service) to calculate an optimal point to give a tradeoff between the key distribution period and the network traffic. But once an entity's master key exposed, the security proof of entire system will be collapsed.

Based on [11], [12] Rezai *et al.* proposed a public key infrastructure ECDH (Elliptic Curve Diffie-Hellman) key exchange solution to generate and exchange session symmetric key with Elliptic Curve algorithm under Diffie-Hellman mechanism. But there is a MITM Attack risk for this solution. If there is an adversary C stay in-between two legal entities A and B, C can intercept the communication between A and B without being identified. So this kind of MITM Attack can not be avoided if the initial shared master key is compromised in the key scheme proposed in [12].

In [13] Lim provided an ID-based key scheme for SCADA system. The additional KDC (Key Distribution Center) or PKG (Private Key Generator) role involvement and another security protection required for the acquiring ID based private key procedure which will increase the communication overhead and introduce new vulnerabilities.

In [14] Lim *et al.* use HMAC for message authentication but without data encryption to decrease the overhead for computing. But the switches, or nodes in the networks through which the data pass are vulnerable to both the intruders or the disgruntled employees. Once the data are declosed the adversaries with knowledge of the procedures can analyze the parameters of the controlled process by analyzing the declosed data. They can derive the abnormal paramaters from the normal ones which can cause the system to fail.

To solve these problems, the vendors should take the responsbility, but in the vendors solutions most equipments should be relaced. To most old plants, replacing all of the IEDs and RTUs is impractical because of the high cost. We need a solution as a transition. So in this paper we provide a solution of a new trusted-ID referenced scheme to secure SCADA communication efficiently with the existing equipments in the plants with the help of sDTU (secure Data Transmission Unit).

A. THE SCADA COMMUNICATION STRUCTURE

In iron and steel plants, the SCADA network communication often uses the three layered structure as Fig. 1 demonstrates.

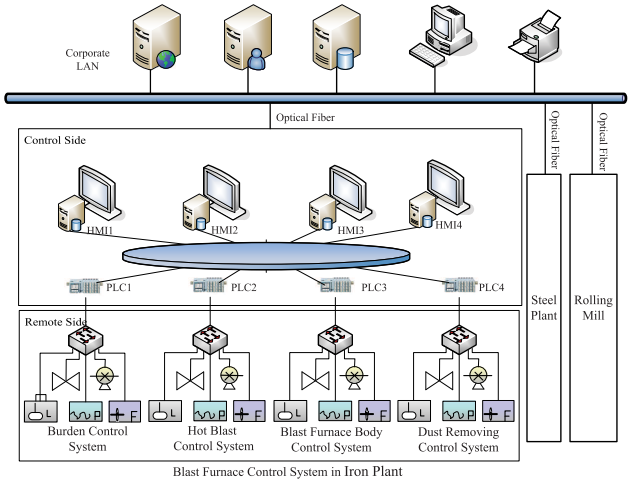


FIGURE 1. Communication structure in iron and steel plants.

The communications in the bottom layer are between the RTUs (Remote Terminal Unit), IEDs (Intelligent Electronic Device) and the master station or the sub master stations. The RTUs or the IEDs in the remote side send the values measured from sensors to the PLCs, and the PLCs send the control instructions to the actuators, e.g. the valves, and simultaneously send the data to the databases of the control stations and HMIs (Human Machine Interface) in the control side.

The communications in the middle layer are between the controllers or the computers of different sub prodecures. The data include the parameters dilivered among defferent sub procedures which will impact the join of them e.g. in Fig. 1, the temprature and pressure from the sensors of Hot Blast subprocedure will be send to PLC2 which control this procedure, and the PLC1 in the Burden procedure will ask these values from PLC2 to make the burden decision. At the same time, these informations will be send to the HMI and the database in the corporate LAN.

The communications in the top layer are between the computers in the corporate LAN, and the data include the production plan and personal informations of administrators and operaters which are vital for the secure of the networks and the control of the whole system.

IPsec and VPNs can be used in the top layer and the middle layer to encrypt the data in the transmission channel. Our solution sDTU is used in the bottom layer to avoid MITM Attack and Replay Attack.

PLCs from different producers such as Seimens, Rockwell and Schneider are used in the above structure. PLCs of different brand use different communication protocols like Ethernet/IP, Profinet, Modbus/TCP. Only few of the protocol above uses encryption, and not with enough security. L. Cheng analyzed the encryption used in S7Commplus in [15] and advised to encrypt the whole packets instead of the key byte encryption.

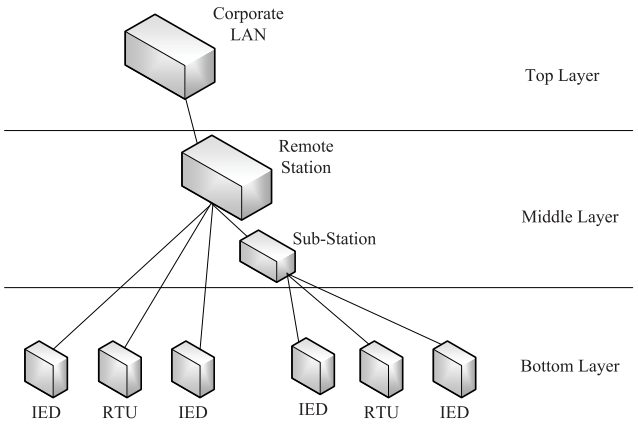


FIGURE 2. Layered SCADA communication architecture.

Make the communication structure of Fig. 2 as the example, two encryption approaches can be used: link encryption and end-to-end encryption.

Link encryption is an approach to communications security by enciphering and deciphering all traffic at each network routing point (e.g. network switch, or node through which it passes) until arrival to its final destination. In most cases there will be several sub-master stations in middle layer to act as a relay which will be the vulnerable points likely to be attacked. Fig.2 depicts this kind of layered architecture.

End-to-end encryption is a system of communication where only the authorized communicating users can understand the messages. In principle, it prevents potential attackers from being able to access the cryptographic keys needed to decrypt the conversation.

In this scenario, to avoid the vulnerability caused by link encryption, end-to-end encryption is adopted for the data to be communicated between master stations and RTUs with Ethernet.

B. CONTRIBUTION

In order to avoid the shortcomings in [9]–[13] and [14], we proposed a new trusted-ID referenced session key scheme solution for end-to-end secure communication.

This solution for securing SCADA communication provides an efficient way to protect the data and protocol between the controller and the RTU, and offers an authentication for the communication, which can avoid MITM Attack and Replay Attack effectively.

In this solution, we make three main contributions:

1) We provide a solution with trusted ID, so that a dynamic trusted ID list can be updated automatically without any other additional procedures. What we discussed is an end-to-end solution, and any devices and terminals in the system can be treated as an independent end without being assigned distinguished roles. Even from application point of view, there are some devices may act as hosts and others as slave terminals, but from our proposed secure communication point of view, they are the same as the communication ends with fixed communication IDs, e.g. network MAC address and/or IP address. By the scheme discussed in this paper, any communication end can setup a secure communication channel automatically with its communication neighbours when it joins the network.
2) This solution is devided into two phases: the offline phase and the online phase. The offline phase is used to achieve the digital signatures of the IDs and the sDTUs, and put the signatures, the IDs and the keys in to the sDTUs. The offline implementation won't increase too much communication overhead. In the online phase the two entities first conduct a mutual authentication by verifying the signature and this is the way to avoid the MITM Attack. If any signature verified invalid, the communication is attacked and the process will stop, otherwise, the process go on to generate random number as the session key. Then verify the session key to avoid the Replay Attack. After the session key is verified to be valid, the process go on to decrypt the data packgaes.

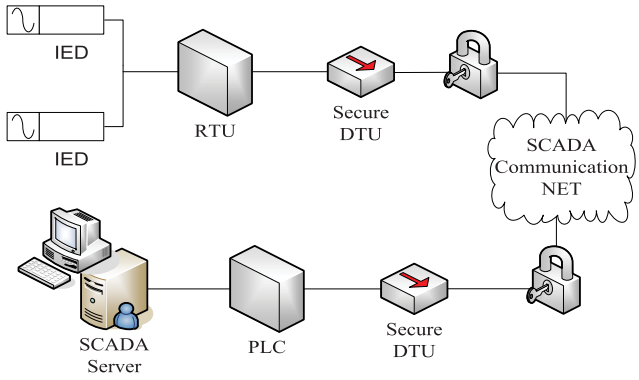


FIGURE 3. The position of the SDTU.

3) With the help of an add-on sDTUs, the sDTUs can be a mountable security module positioned as Fig.3 illustrates in a general communication environment or a security chip that can be embedded in the IEDs and RTUs. The form of sDTU depends on the demand of the system. All of the security information should be stored

inside the embedded anti-tamper secure chip without exposing to outside, and other critical information such as the paired IDs' information list, can only be modified with authorization.

The rest of the paper is organized as follows. Section II is a review of SCADA key schemes. Section III is a discussion of the proposed scheme in detail, a proof for the security of the scheme and an example are offered. Section IV is the analysis for the performance. Section V is the conclusion and future studies.

II. A REVIEW OF SCADA KEY SCHEMES

A. BACKGROUND ON CRYPTOSYSTEMS

The AES (Advanced Encryption Standard) algorithm is one of the most widely used symmetric key block cipher algorithm to encrypt/ decrypt blocked data being transmitted between the master side and the remote side in the SCADA system because of its fast calculating speed and high security, so we use it in the encryption and decryption of the communicating data after session key established.

Apart from symmetric algorithm, public key cryptosystems perform encryption and decryption in an asymmetric way. The traditional public key cryptosystem is RSA which is based on a hard problem of factoring large numbers.

ECC (Elliptic Curve Cryptography) [16], [17] is an efficient alternative to RSA. Compared with RSA, the major advantage of ECC is that it offers comparable security with smaller key size. ECCDSA (Elliptic Curve Cryptography Digital Signature Algorithm) is the digital signature algorithm based on ECC. For current and future security levels ECCDSA offers better performance values than RSA based signatures.

In ID based cryptography [18] a user can choose an arbitrary string as its public key. An IP address or MAC address can become a user agent's public key. A trusted third party, named as the PKG, will generate a private key based on the public key of the requesting user. Using its public key a receiver can authenticate the sender by validating the ID based digital signature, which was generated with the sender's private key.

The major advantage of ID based cryptosystems is that no PKI (Public Key Infrastructure) is needed for key management. Compared with other cryptosystems ID based cryptosystems are quite secure.

Most SCADA secure communication proposals use the session key to encrypt the data transmitting between the SCADA entities, i.e. from master station to bottom layer PLCs, RTUs and IEDs, or first to sub-master station and then forward to bottom layer devices.

And the three kinds of session key schemes mentioned previously will be reviewed hereafter.

B. PRELOADED SYMMETRIC MASTER KEY

In order to generate a temporary session key, there must be a preloaded master key existing in each communication entitiy, let's assume A and B as the two SCADA entities need to setup a secure communication channel. And K_m is the preloaded master key shared by A and B. Either A or B can be acted as the session initiator to start the procedure of session key generating. We can suppose A start the key generating procedure, the related steps described as below:

Step 1:

A: Generates a random number RND_A;

Step 2:

A: Ciphers RND_A with predefined algorithm and uses K_m as the encryption key and output $\{RND_A\}_{K_m}$;

Step 3:

$A \Rightarrow B$: $\{RND_A\}_{K_m}$;

Step 4:

B: Decrypts $\{RND_A\}_{K_m}$ with the same algorithm and the same master key shared with A, so as to recover $RND_A = \{\{RND_A\}_{K_m}\}_{-K_m}$;

Step 5:

Both A and B use RND_A as the temporary session key to securely communicate with each other. But for this solution, once an entity's master key exposed, the security proof of entire system will be collapsed.

Even there are some variants in different preloaded master key proposals, but the basic processes are almost alike.

C. ID-BASED KEY SCHEME

The ID-based key scheme uses any bi-linear map $\tilde{e} : G_1 \times G_1 \rightarrow G_2$, G_1 and G_2 are cyclic groups of the same order. Between 2 groups, G_1 and G_2 as long as a variant of Computational Deffie-Hellman problem in G_1 is difficult. This meets the requirement of Shamir who asked a public key encryption scheme in which the public key can be an arbitrary string.

As Dan Boneh et al. discussed in [19] Weil paring on elliptic curve is one of the bi-linear map can be used for the ID-based encryption.

Unlike certificate-based PKI which requires a CA (Certificate Authority) , the ID-based scheme needs a KGC (Key Generation Center), also called PKG with function of generating secure communication entities' private key from their ID strings. But this will increase the communication overhead and introduce new vulnerabilities.

Reference [20] defines some other kinds of bilinear map parings over elliptic curve to achieve ID-based algorithms.

D. ECDH KEY SCHEME

The ECDH key scheme has been studied for a long time, and already is accepted as the ISO/IEC standards.

If there are two entities A and B need to share a secret value, they first find an elliptic curve $E(F_p)$, and each generates a public key pair (K_{priA}, K_{pubA}) and (K_{priB}, K_{pubB}), both K_{priA} and K_{priB} are randomly selected in the range of $[1, n]$ and should be kept secret by each other, while K_{pubA} and K_{pubB} are used as public key and calculated as:

$$K_{pubA} = K_{priA} \times g$$
$$K_{pubB} = K_{priB} \times g$$

where g is the base point of $E(F_p)$.

Then the shared secret value between A and B can be produced using formula as denoted below:

$$\begin{aligned} K_{priAB} &= K_{priA} \times K_{pubB} \\ &= K_{priB} \times K_{pubA} \\ &= K_{priA} \times K_{priB} \times g \end{aligned}$$

Since K_{priA} and K_{priB} are secretly kept by A and B, nobody else can obtain the value of K_{priAB} without knowing either K_{priA} or K_{priB}.

If there is an adversary C stay between two legal entities A and B, the previously discussed shared secret $K_{priA} \times K_{priB} \times g$ will be replaced by C as: $K_{priA} \times K_{priC} \times g$ as a shared secret between A and C, and $K_{priC} \times K_{priB} \times g$ as another shared secret shared between C and B. This kind of MITM Attack can not be avoided.

III. THE PROPOSED SCHEME

This paper proposed a procedure plus an ECCDSA algorithm implementation to achieve a high-performance and secured session key generation scheme for SCADA communication protection.

Two phases have been defined in this proposal: the offline phase and the online phase, and each phase will conduct several steps in order to complete the whole procedure.

A MSK (Master Signing Key) will be generated first, and is used for signing identity and public key information of each sDTU.

The key pair of each sDTU with a reference of entity's ID are also generated and loaded into the SM(Secure Module) in each sDTU together with the public key of MSK.

A. ELLIPTIC CURVE CRYPTOGRAPHY DIGITAL SIGNATURE ALGORITHM

The security keystone of ECC is based on the difficulty of solving DLP (Discrete Logarithm Problem) on EC (Elliptic Curve) over finite field F_p.

The EC over finite field F_p denoted as: $E(F_p)$ defines a set of n points (n is called the order of the curve) satisfying equation:

$$(Y^2 = X^3 + aX + b) mod\ p$$

where p is a prime number and with constrains:

$$(4a^3 + 27b^2 \neq 0) mod\ p$$

The analog of modular exponentiation is the point multiplication operation where the point addition operation is performed as many times as the multiplier value.

When P is a point on the EC, k is the number of Ps to be added up, the sum is represented as kP, the ECDLP (Elliptic Curve Discrete Logarithm Problem) problem is to find the unknown k from P and kP.

Among the points of $E(F_p)$, a base point g can be found so that all of the other $(n-1)$ points among this EC points set can be calculated by g with different times of self addition :

$$\forall P \in E(F_p), \exists k \in [1, n] \Rightarrow k \times g = P \tag{1}$$

So P can be exposed publicly, and k must be kept privately, they are a pair of cryptographic keys to encrypt information or to sign a digital signature with related procedures.

ECCDSA is to verify a signature result with the public key to prove the result is really generated by the related private key with ECC algorithm.

The digital signature generating process:

(1) Generate a random private key as K_{pri}, $g(x_g, y_g)$ is the base point of the EC, generate a message digest m_d with Hash;

(2) From (1) we can get:

$$K_{Apub}(X_A, Y_A) = K_{pri} \times g(x_g, y_g) \tag{2}$$

The key pair of A is $(K_{Apri}, K_{Apub}(X_A, Y_A))$, based on elliptic curve $E(F_p)$ with the order of n;

(3) Randomly select a k in the range of $[1, n]$, calculate (s, r), if $s = 0$, reselect k and calaulate again:

$$\begin{cases} r = X_A\ mod\ n \\ s = k^{-1}(m_d + rK_{Apri})\ mod\ n \end{cases} \tag{3}$$

(4) (r, s) is the signature;

The signature verification process:

(1) The same message digest m_d will be generated;

(2) Calculate:

$$\begin{cases} w = s^{-1}\ mod\ n \\ u_1 = m_d w\ mod\ n, \quad u_2 = rw\ mod\ n; \end{cases} \tag{4}$$

(3) Calculate curve point:

$$K'_{Apub}(X'_A, Y'_A) = u_1 \times g + u_2 \times K_{Apub} \tag{5}$$

(4) If $r \equiv X'_A \bmod n$, the signature is valid.

The signature is correct because:

$$\begin{cases} K'_{Apub} = u_1 \times g + u_2 \times K_{Apub} \\ K'_{Apub} = m_d w \times g + rw \times K_{Apub} \\ K'_{Apub} = m_d w \times g + rwK_{Apri} \times g \\ K'_{Apub} = (m_d + rK_{Apri})s^{-1} \times g \\ K'_{Apub} = (m_d + rK_{Apri})(m_d + rK_{Apri})^{-1}(k^{-1})^{-1} \times g \\ K'_{Apub} = k \times g = K_{Apub} \end{cases}$$

B. THE OFFLINE PHASE

1) MASTER SIGNING KEY

The MSK pair is generated under ECC algorithm by the HSM (Hardware Security Module). From (1), (2), (3), the public part MSK_{pub} will be self-signed with its private part MSK_{pri} to get the signature $\{H_{MSKpub}\}_{-MSK_{pri}} = (r_{MSK}, s_{MSK})$:

$$\begin{cases} MSK_{pub}(X_P, Y_P) = MSK_{pri} \times g(x_g, y_g) \\ m_{dMSK} = H_{MSKpub} = Hash(MSK_{pub}) \\ r_{MSK} = X_P\ mod\ n \\ s_{MSK} = k^{-1}(H_{MSKpub} + rMSK_{Apri})\ mod\ n \end{cases} \tag{6}$$

2) ID ASSIGNMENT AND SIGNATURE

Each sDTU will be assigned a string as the unique ID sequence e.g. use the entity's MAC and IP address.

Then we can calculate the Hash of this ID:

$$H_{ID} = Hash(ID) \tag{7}$$

Generate a random private key for sDTU as K_{pri}, $g(x_g, y_g)$ is the base point of NIST 192, from (2) we can get:

$$\begin{cases} K_{pub1}(X_{P1}, Y_{P1}) = H_{ID} \times g(x_g, y_g) \\ K_{pub2}(X_{P2}, Y_{P2}) = K_{pri} \times K_{pub1}(X_{P1}, Y_{P1}) \end{cases} \tag{8}$$

K_{pri} is the private key of sDTU, K_{pub1} and K_{pub2} are the related public key.

Calculate the Hash of the data sequence $ID \| K_{pub2}$, we get $m_{d\,ID} = H_{ID \| K_{pub2}}$.

In NIST 192, $n = FFFFFFFFFFFFFFFFFFFFFFFF99DEF836146BC9B1B4D22831$, from (3) we can sign $H_{ID \| K_{pub2}}$ with MSK_{pri} to get the signature $\{H_{(ID \| K_{pub2})}\}_{-MSK_{pri}} = (r_{ID}, s_{ID})$:

$$\begin{cases} r_{ID} = X_{P2} \bmod n \\ s_{ID} = k^{-1}(H_{ID \| K_{pub2}} + rMSK_{pri}) \bmod n \end{cases} \tag{9}$$

At last load the $\{H_{MSKpub}\}_{-MSK_{pri}}$, K_{pri}, ID, K_{pub2}, $\{H_{(ID \| K_{pub2})}\}_{-MSK_{pri}}$, MSK_{pub} into the anti-temper SM embedded in the sDTU so as to assure K_{pri} being kept secretly, and all of other value can not be modified without authorized permission. The process is shown in Fig. 4.

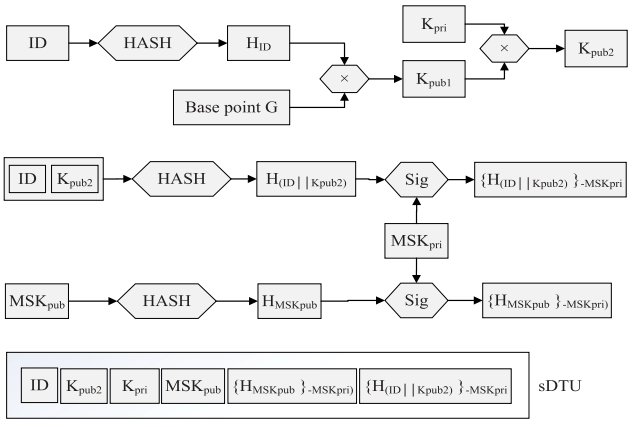


FIGURE 4. The offline phase.

Remark 1: The offline phase is used to achieve the digital signatures of the IDs and the sDTUs, and put the signatures, the IDs and the keys into the sDTUs. The offline implementation won't increase too much communication overhead.

C. THE ONLINE PHASE

When the two sDTUs need to communicate with each other, one of them will initiate the session and start the authentication process before a secure communication channel can be established as shown in Fig. 5.

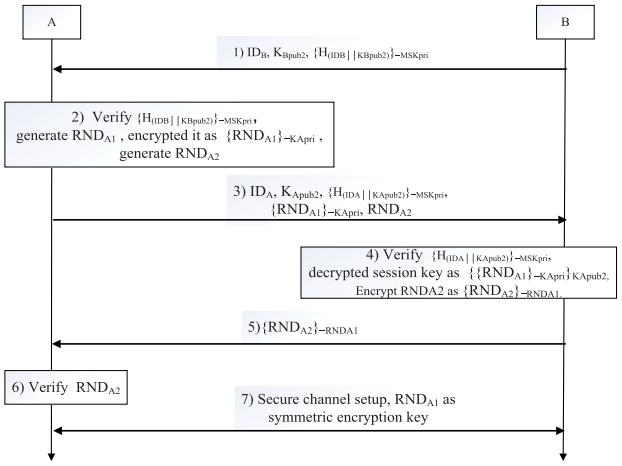


FIGURE 5. The online phase.

1) INITIATE

$B \longrightarrow A : ID_B, K_{Bpub2}, \{H_{(ID_B \| K_{Bpub2})}\}_{-MSK_{pri}}$.

Assume sDTU A and sDTU B will communicate as the secure channel initiator. B will send its ID_B, public key K_{Bpub2}, as well as the digital signature $\{H_{(ID_B \| K_{Bpub2})}\}_{-MSK_{pri}}$ to A.

2) AUTHENTICATION

From (4), (5), (6), the signature of $\{H_{MSKpub}\}_{-MSKpri}$ will first be verified with MSK_{pub}. The process is:

$$\begin{cases} w = \{k^{-1}(H_{MSKpub} + rMSK_{pri}) \bmod n\}^{-1} \bmod n \\ u_1 = (H_{MSKpub})w \bmod n \\ u_2 = (X_P \bmod n)w \bmod n \\ K'_{pub}(X'_P, Y'_P) = u_1 \times g + u_2 \times MSK_{pub} \end{cases} \tag{10}$$

If $r_{MSK} \equiv X'_P \bmod n$, the signature is valid.

Then A and B will conduct a mutual authentication to verify the signature of $\{H_{(ID_A \| K_{Apub2})}\}_{-MSK_{pri}}$ and $\{H_{(ID_B \| K_{Bpub2})}\}_{-MSK_{pri}}$ respectively with MSK_{pub}. If any signature verified invalid, the process will stop, and secure communication will abort.

3) SESSION KEY GENERATION

(1) A will verify $\{H_{(ID_B \| K_{Bpub2})}\}_{-MSK_{pri}}$ by (10), generate a random number RND_{A1} as session key, encrypt it with K_{Apri}, K_{Bpub1} and K_{Bpub2} as C_2.

$$\begin{cases} C_1 = K_{Apri} \times K_{Bpub1} \\ C_2 = RND_{A1} + K_{Apri} \times K_{Bpub2} \end{cases} \tag{11}$$

Then generate another random number RND_{A2}.

(2) $A \longrightarrow B$: $\{H_{(ID_A \| K_{Apub2})}\}_{-MSK_{pri}}$, K_{Apub2}, ID_A, C_2, and RND_{A2}.

1) B will verify $\{H_{(ID_A \| K_{Apub2})}\}_{-MSK_{pri}}$ by (10), decrypt RND_{A1} with (11) and (12):

$$\begin{cases} K_{Bpub2} = K_{Bpri} \times K_{Bpub1} \\ RND_{A1} = C_2 - K_{Bpri} \times C_1 \end{cases} \tag{12}$$

Then RND_{A1} as the key encrypt RND_{A2} with symmetric encryption algorithm.

(3) $B \longrightarrow A$: $\{RND_{A2}\}_{-RND_{A1}}$.
(4) A will decrypt RND_{A2} with RND_{A1} to verify RND_{A2}.
(5) Secure channel setup with RND_{A1} as the session key to establish secure communication channel protected with symmetric encryption algorithms.

Remark 2: The verification of the signature can avoid the MITM Attack and the verification of RND_{A1} and RND_{A2} can avoid the Replay Attack.

4) ENCRYPTION WITH SESSION KEY

Once the session key has been shared successfully, the end-to-end secure communication channel will be set up.

In order to assure the integrity and confidentiality communication requirement as well as counterfeiting the replay attack, the CBC-MAC(Cipher-Block Chaining mode Message Authentication Code) calculation should be implemented.

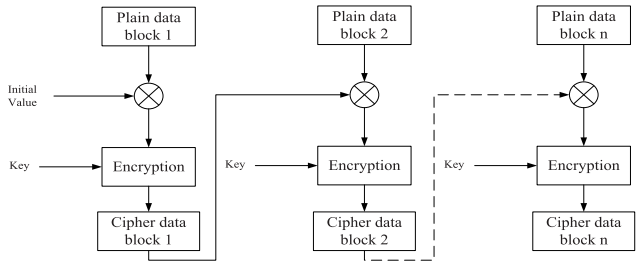


FIGURE 6. The CBC-MAC calculation.

The encryption process in Fig. 6 is as:

(1) The sender generates a random number as initial value;
(2) Split the plaintext into several blocks, each block meets the size request of the symmetric encryption algorithm, e.g. in AES algorithm the block size can be 16 bytes;
(3) No matter the last block size is exactly 16 bytes or not there should be a mandatory padding appended, the total overhead is at most 2 blocks;
(4) Exclusive or the initial value with the first block of data, and then encrypt the result with key output the first encrypted block;
(5) Then exclusive or the first block output with the second block of plaintext and encrypt again, and continue till the last block;
(6) Since the last block should contain a patterned padding bytes, e.g. 0x80, 0x00, 0x00...0x00, if the receiver decrypts the massage without finding this special padding, then there must be some unauthorized modification of the data, and the data should be rejected. This acts as the MAC.

Remark 3: The encryption can meet confidential requirement and CBC-MAC calculation for data integrity approach can be achieved.

D. MULTI-ENDS CONSIDERATION

In the above sections we discussed the general end-to-end solution in details, and we assumed that each end is managed by the same master station, but they can communicate with each other without visiting the master station any more after the offline procedure.

If we consider the one-to-multi scenario, such as one master station communicates with several remote terminals. The master station can act as entity A and any other remote terminals which need to communicate with this master station should act as the different entity Bs, e.g. $B_1, B_2, \ldots\ldots, B_n$. In this way the master station is one of the communication ends, and its asymmetric key pair will not be treated as the authority to sign communication ends' ID plus public key as described above. But the master station's ID and public key together with the mentioned different entity Bs' ID and public key should be signed by an up level system asymmetric key pair as the authority. And in this scenario, the so-called master station can manage a trusted ID list after it complete the above discussed key scheme process. Since each communication channel will generate its own session key, the master station can index each session key with their trusted IDs managed by itself.

E. SECURITY PROOF

BAN(Burrows-Abadi-Needham) Logic [21] is widely used to analyze the completeness of protocols. GNY (Gong-Needham-Yahalom) logic [22] is the extension over BAN and overcomes BAN's limitation. So GNY logic is adopted here to analyze the security of the proposed key scheme. Firstly, some formulae and statements used in the GNY logic will be introduced, then the goals and assumptions will be set to prove the key scheme is valid by GNY logic.

1) FORMULAE AND STATEMENTS

In the GNY logic, a formula is a name used to indicate a bit string, with a particular value in a run [22]. In order to describe the GNY logic, first let symbols X and Y range over formulae. Then, let's introduce some formulae used in the key scheme protocol proof, for the complete list of all logical postulates is described in [22].

(1) (X, Y): conjunction of two formulae X and Y.
(2) $\{X\}_K$ and $\{X\}_K^{-1}$: symmetrically encrypt and decrypt X with the key K.
(3) $\{X\}_{+K}$ and $\{X\}_{-K}$: asymmetrically encrypt and decrypt X with the public key $+K$ and the private key $-K$.
(4) $H(X)$: a one-way Hash function of X.
(5) $F(X_1, \ldots X_n)$: is a many-to-one computationally feasible function.
(6) $*X$: X is not originated here.
(7) $\{X\}_{+K_{Bpub2}, -K_{Apri}}$ and $\{X\}_{-K_{Bpri}, +P_{Apub2}}$ asymmetrically encrypt X with principal B's public key K_{Bpub2} and principal A's private key K_{Apri}, and decrypt X with principal B's private key K_{Bpri} and principal A's public key P_{Apub2}.

Additionally, another formula based on the proposed scheme will defined, and the similar statements should be assigned as with the $+K/-K$ asymmetrically encrypt and decrypt.

A basic statement reflects some property of a formula. Let symbols P and Q be principals. The followings are statements used in our proof.

(1) $P \lhd X$: P is told formula X.
(2) $P \ni X$: P possesses formula X.
(3) $P \mid\sim X$: P once conveyed formula X.
(4) $P \mid\equiv \sharp(X)$: P believes that X is fresh.
(5) $P \mid\equiv \phi(X)$: P believes that X is recognizable.
(6) $P \mid\equiv P \overset{S}{\leftrightarrow} Q$: P believes that S is a suitable secret for P and Q.
(7) $P \mid\Rightarrow X$: P has jurisdiction over X.
(8) $P \lhd *X$: P is told that a formula X which did not convey previously in the current run.
(9) $P \mid\equiv \overset{+K}{\longmapsto} Q$: P believes that $+K$ is a suitable public key of Q.

2) PROTOCOL DESCRIPTIONS AND GOALS

In order to fit the GNY logic, some notations are changed and the proposed key scheme protocol is transformed into the form of $P \longrightarrow Q : (X)$. The MSK's private key is denoted as $-K$, and the corresponding public key is denoted as $+K$. Assume entity A as the secure channel setup initiator, A will generate two random numbers, one is denoted as RND_A for future shared key, the other is N_A as to generate the confirmation ticket from B. Private key of A is denoted as $-K_{Apri}$, its related public key is denoted as $+K_{Apub}$, private key of B is denoted as $-K_{Bpri}$, its related public key is denoted as $+K_{Bpub}$.

(1) $B \longrightarrow A : (\{ID_B, K_{Bpub2}, \{H_{(ID\|K_{Bpub2})}\}_{-K})$
1) $A \longrightarrow B : (ID_A, K_{Apub2}, \{H_{(ID\|K_{Apub2})}\}_{-K}, \{RND_A\}_{+K_{Bpub2},-K_{Apri}}, N_A)$
(2) $B \longrightarrow A : (\{N_A\}_{RND_A})$

Next, we describe the goals in the key scheme protocol.

a: MESSAGE CONTENT AUTHENTICATION

Goal 1: A believes the message in the first run is recognizable.

$$A \mid\equiv \phi(\{ID_B, K_{Bpub2}, \{H(ID_B||K_{Bpub2})\}_{-K}\})$$

Goal 2: B believes the message in the second run is recognizable.

$$B \mid\equiv \phi(ID_A, K_{Apub2}, \{H(ID_A||K_{Apub2})\}_{-K}, \{RND_A\}_{+K_{Bpub2},-K_{Apri}}, N_A)$$

b: MESSAGE ORIGIN AUTHENTICATION

Goal 3: B believes A conveyed the message in the second run.

$$B \mid\equiv A \mid\sim \{RND_A\}_{+K_{Bpub2},-K_{Apri}}$$

Goal 4: A believes B conveyed the message in the third run.

$$A \mid\equiv B \mid\sim (\{N_A\}_{RND_A})$$

c: SESSION KEY ESTABLISHMENT

Goal 5: A believes that B possesses RND_A.

$$A \mid\equiv B \ni RND_A$$

3) ASSUMPTION LIST

Reference [22] gives out some logical postulates:

$$T_1 : \frac{P \lhd *X}{P \lhd X} \tag{13}$$

$$T_6 : \frac{P \lhd \{X\}_{-K}, P \ni +K}{P \lhd X} \tag{14}$$

$$P_1 : \frac{P \lhd X}{P \ni X} \tag{15}$$

$$R_1 : \frac{P \mid\equiv \phi(X)}{P \mid\equiv \phi(X, Y), P \mid\equiv \phi(F(X))} \tag{16}$$

$$R_4 : \frac{P \mid\equiv \phi(X), P \ni -K}{P \mid\equiv \phi(\{X\}_{-K})} \tag{17}$$

$$R_6 : \frac{P \ni H(X)}{P \mid\equiv \phi(X)} \tag{18}$$

According to the postulates above, some assumptions are made as follow:

1. Secret key RND_A and nonce N_A are generated by A in the proposed protocol, so A possesses RND_A and N_A and believes they are fresh and recognizable, A also possesses the private key $-K_{Apri}$ and the public key $+K_{Apub2}$, and MS's public key $+K$, since they are stored in A, besides A generates RND_A as the session key, so A believes that RND_A is the secret share between A and B.

$$I_1 : \frac{P \lhd *\{X\}_K, P \ni K, P \mid\equiv P \overset{K}{\leftrightarrow} Q, P \mid\equiv \phi(X), P \mid\equiv \sharp(X, K)}{P \mid\equiv Q \mid\sim X, P \mid\equiv Q \mid\sim \{X\}_K, P \mid\equiv Q \ni K} \tag{19}$$

$$I_4 : \frac{P \lhd \{X\}_{-K}, P \ni +K, P \mid\equiv P \overset{+K}{\longmapsto} Q, P \mid\equiv \phi(X)}{P \mid\equiv Q \mid\sim X, P \mid\equiv Q \mid\sim \{X\}_{-K}} \tag{20}$$

$$\frac{A \lhd *(ID_B, K_{Bpub2}, \{H(ID_B||K_{Bpub2})\}_{-K}), A \ni +K}{A \lhd ID_B, A \lhd K_{Bpub2}, A \lhd \{H(ID_B||K_{Bpub2})\}_{-K}, A \ni H(ID_B, K_{Bpub2})} \tag{21}$$

$$\frac{A \ni H(ID_B||K_{Bpub2})}{A \mid\equiv \phi(ID_B||K_{Bpub2}), A \mid\equiv \phi(\{ID_B, K_{Bpub2}, \{H(ID_B||K_{Bpub2})\}_{-K}\})} \tag{22}$$

$A \ni RND_A, A \ni N_A, A \ni -K_{Apri}, A \ni +K_{Apub2}, A \ni +K, A \mid\equiv \sharp(RND_A), A \mid\equiv \sharp(N_A), A \mid\equiv \phi(RND_A), A \mid\equiv \phi(N_A), A \mid\equiv A \overset{RND_A}{\longleftrightarrow} B$

2. B possesses $-K_{Bpub2}$, $+K_{Bpri}$, as well as MS's public key $+K$, since they are stored in B.

$$B \ni -K_{Bpri}, B \ni +K_{Bpub2}, B \ni +K$$

4) AUTHENTICATION PROOF USING GNY LOGIC

Use GNY logic to analyze our protocol. A complete list of all logical postulates and the index in the list is provided in [22] to show how to achieve the goals.

a: THE FIRST RUN

From (13),(14),(15), we can get (21).

Since A possesses the MS's public key $+K$, and is told $(\{ID_B, K_{Bpub2}, \{H(ID_B||K_{Bpub2})\}_{-K}\})$, thus the component $\{H(ID_B||K_{Bpub2})\}_{-K}\}$ is also told, so we can conclude A possesses $\{H(ID_B||K_{Bpub2})\}$.

From (16),(18), we can get (22).

Since A possesses the Hash $H(ID_B||K_{Bpub2})$, then A is entitled to believe that $(ID_B||K_{Bpub2})$ is recognizable, therefore A also believes

$$(\{ID_B, K_{Bpub2}, \{H(ID_B||B_{Bpub2})\}_{-K}\})$$

is recognizable. (Goal 1)

b: THE SECOND RUN

Since B possesses MS's public key $+K$ as A does, so we can also conclude that B believes:

$(\{ID_A, K_{Apub2}, \{H(ID_A||K_{Apub2})\}_{-K}, \{RND_A\}_{+K_{Bpub2},-K_{Apri}}, \{N_A\}_{RND_A}\})$ is recognizable in the same way. (Goal 2)

$$\frac{B \mid\equiv MS \Rightarrow \overset{K_Apub2}{\longmapsto} A, B \mid\equiv MS \mid\equiv \overset{K_Apub2}{\longmapsto} A}{B \mid\equiv \overset{+K_{Apub2}}{\longmapsto} A} \quad (23)$$

Since MS is considered as the authority over both A and B's public key signing, according Jurisdiction Rules (23), we can know that: $B \mid\equiv \overset{+K_{Apub2}}{\longmapsto} A$.

From (17),(20), we can get (24), as shown at the bottom of this page.

If all of the following four conditions hold:

i) B receives a formula RND_A, encrypted with A's private key and B's public key;

ii) B possesses the corresponding A's public key and B's private key;

iii) B believes the public key is A's;

iv) P believes RND_A is recognizable.

Then B is entitled to believe:

i) A once conveyed the formula RND_A;

ii) A once conveyed the formula consisting RND_A encrypted with private key K_{Apri} and public key K_{Bpub2}. (Goal 3)

c: THE THIRD RUN

From (19), we can get (25), as shown at the bottom of this page.

If all of the following conditions hold:

i) A receives a formula consisting with N_A encrypted with key RND_A, and marked not originated here;

ii) A possesses RND_A;

iii) A believes that RND_A is a suitable secret for him and B;

iv) A believes formula N_A is recognizable;

v) A believes that RND_A is fresh, or that N_A is fresh.

Then A is entitled to believe:

i) B once conveyed N_A;

ii) B once conveyed N_A encrypted with key RND_A;

iii) B owns key RND_A. (Goal 4) (Goal 5)

F. AN EXAMPLE FOR DEMOSTRATION

1) ID AND MSK GENERATION

Suppose there are two entities connected inside the SCADA network, with different IP and MAC address. SHA1(Other data digest algorithms such as SHA2, SHA3 can also be used) is used to generate the Hash value, their ID string and Hash value can be represented from (7) as shown in Table 1.

TABLE 1. ID and MSK generation.

Entity A	IP_A:192.168.1.10 MAC_A: 00:1B:1B:90:6D:45 ID_A : C0A8010A 001B1B90 6D45 H_A :80BC3506 6D3F11E8 B7DB6BAB B9C9FDC6 20207411
Entity B	IP_B:192.168.1.15 MAC_B: 00:1B:1B:96:55:61 ID_B : C0A8010F 001B1B96 5561 H_B :E25BFABF D61A08A9 DB547A69 BA4964E7 89FE81CE

We use NIST-192 curve to generate MSK key pair. Firstly we generate a random MSK private key MSK_{pri}, then calculate $MSK_{pri} \times G$ obtain the public key point MSK_{pub} (X_p, Y_P) shown in Table 2.

$$\frac{B \lhd (\{RND_A\}_{+K_{Bpub2},-K_{Apri}}), B \ni (-K_{Bpri}, +K_{Apub2}), B \mid\equiv \overset{+K_{Apub2}}{\longmapsto} A, B \mid\equiv \phi(RND_A)}{B \mid\equiv A \mid\sim RND_A, B \mid\equiv A \mid\sim (\{RND_A\}_{+K_{Bpub2},-K_{Apri}})} \quad (24)$$

$$\frac{A \lhd *\{N_A\}_{RND_A}, A \ni RND_A, A \mid\equiv A \overset{RND_A}{\longleftrightarrow} B, A \mid\equiv \phi(N_A), A \mid\equiv \sharp(RND_A, N_A)}{A \mid\equiv B \mid\sim (N_A), A \mid\equiv B \mid\sim \{N_A\}_{RND_A}, A \mid\equiv B \ni (RND_A)} \quad (25)$$

TABLE 2. Point G and MSK key.

Point G	X_g:188DA80E B03090F6 7CBF20E B43A1880 0F4FF0AF D82FF1012 Y_g:07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811
MSK_{pri}	577E6125 2C73C22B 3B1223EF C6674B8D 1AD0E7C6 D629D097
MSK_{pub}	X_P : CE739AB4 85AC6FB1 2B6807B8 9EA76D05 3F24E7E2 18F23B87 Y_P : FD04B06B 1D7C2415 73E6C0CC 172603FF B1CBB6FF D8D6AF58

2) ENTITY A

Next we generate a random private key K_{Apri}, from (8) we can obtain point (X_{PA1}, Y_{PA1}) and point (X_{PA2}, Y_{PA2}) as shown in Table 3.

TABLE 3. Entity A.

K_{Apri}	F4369064 6A50ACC9 BD5F488F 8539AC4B 0A04050A 5F55FE19
K_{Apub1}	X_{Apub1} :447965EB 8BA7DAC7 70CDB287 3ABE4650 69EE82C6 E2A17C74 Y_{Apub1} :2B3D23AD D9FD4EE1 D4DC43FE 2FD9223A 35994F72 17F2E9BE
K_{Apub2}	X_{Apub2} : 89785369 AD59F155 4CBAC896 F63F0557 1843BA4D 7E01EB7F Y_{Apub2} : DC281F50 522B2C3E F0335738 48D181FF E22E4E78 CF5BED35

3) ENTITY B

The same process with entity A as shown in Table 4.

TABLE 4. Entity B.

K_{Bpri}	DC9B1555 1DE23F07 1B832CA4 820A7261 D0E440D9 D312EF32
K_{Bpub1}	X_{Bpub1} : 27D116B2 917E004C EE1E587B A24AA7BF D07ACDE9 3AAE67DF Y_{Bpub1} : 5B2E801A 5592D2B9 D58ABECA DBABF646 EF182B4B F9D987EA
K_{Bpub2}	X_{Bpub2} :9C914491 1C57CC8C 9BF5632A FE68656A C8988839 3507A9A1 Y_{Bpub2} :D092776E 172269D6 49FF83E3 B0CA8104 40709E37 28BB6ED2

4) DIGITAL SIGNATURE

We calculate $Hash(ID_A, X_{PA2})$ and $Hash(ID_B, X_{PB2})$, then signature with MSK_{pri} respectively by (9) as shown in Table 5.

TABLE 5. Digital signature.

K_{Bpri}	DC9B1555 1DE23F07 1B832CA4 820A7261 D0E440D9 D312EF32
Entity A	$H(ID_A, X_{PA2})$: FD9A2F15 DC4582BC B3637804 1F15D934 A94CD773 R_A : 9F8C0B58 CA7729EE 4BA0704F 11EE5D88 6DC037D3 E3C3F7C0 S_A : 9AE0D3DB 5071EBEB 2BE052A1 6E676765 301C9112 04ACB16B
Entity B	$H(ID_B, X_{PB2})$: A3A26E78 B8789E2B 47B56E8E 6EC28DAF E46B9489 R_B : 17120513 A836B6A7 548B9A6B 40870C33 D63703FB EE3A46 S_B : 37ED88E9 6FCA4FE8 57D1929F 78E03CB7 C6105636 7252DDF0

From (4), (5), (9), (10) the signature of entity A and entity B can be verified.

5) SESSION KEY GENERATION

Entity A produces a random number as the will be shared symmetric key, the public key of A and B, K_{Apub1}, K_{Apub2}, K_{Bpub1}, K_{Bpub2}, are known each other. From (10),(11) we can calculate C_1 and C_2' as shown in Table 6.

Add RND_A onto either X'_{C2} or Y'_{C2} to get $C_2(X_{C2}, Y_{C2})$.

In order to check the decryption, calculate $K_{Bpri} \times C_1$, and the same result can be gotten as point (X'_{C2}, Y'_{C2}). From (12), substrate either X'_{C2} or Y'_{C2} with which RND_A has been previously added, the RND_A will be correctly recovered and be used as the session key to encrypt the data.

TABLE 6. Session key generation.

RND_A	8D37B490 E382C323 2F452335 CA5B5F0A
C_1	X_{C1} :EAA307D7 4CE6D18E AE36854C 0882BACB F1C95D6D 4C7B210F Y_{C1} : 3BA08262 8AD731FC C911B2F0 A5AB366D 538DFBB8 A154B7B0
C_2'	X'_{C2} :E76B5D88 D1F6C28E A7BD7FCD 94C9D738 D1DC39C1 67B1CED1 Y'_{C2} : 660E537F A797A824 DA1EE02F C73C9FE4 86B135E2 23EE80CF

6) ATTACK RESISTING

Scenario I: If an adversary C pretends to be B stay in the middle of A and B, when it tries to recover the RND_A, it can only use a guessed K'_{Bpri} e.g. even only 1 bit different from the real K_{Bpri}, assume the guessed as shown in Table 7.

TABLE 7. Attack I.

K'_{Bpri}	DC9B1555 1DE23F07 1B832CA4 820A7261 D0E440D9 D312EF31
C_2''	X''_{C2} : C2846865 EFFA37FA E52DC94C 45F2B341 BEFEAA61 8E4DA058 Y''_{C2} : 2C96AE87 6C75DD65 3C70555C 5C594055 4C2FE296 E46F1D99

After calculating $K'_{Bpri} \times C_1$, the point (X''_{C2}, Y''_{C2}) is gotten, totally different from the correct value, thus the real shared secret RND_A can not be recovered by C. That is to say, C can not decrypt the data, the failure of MITM attack.

Scenario II: If an entity B' with different ID wants to replace B, even the IP, MAC (192.168.1.22, 00:1B:1B:96:52:33) address could be configured in A's white-list. When A verifies the signature the process as shown in Table 8.

TABLE 8. Attack II.

Entity B'	IP'_B : 192.168.1.22 MAC'_B : 00:1B:1B:90:6D:45 ID'_B : C0A80116 001B1B96 5233 H'_B : 0CC2F95F 2EA2892F C462642C 774D39F6 62F09DAB
$H(ID_{B'}, X_{Bpub2})$	18CE8196 E427794B 7E6A3CF7 0E6DCC9A BBB4B58C

Uses $ID_{B'}$, X_{PB2}, R_B, S_B, $MSK_{pub}(X_p, Y_p)$ as input parameters, and the signature verification result should be invalid.

IV. ANALYSIS

A. COMPARISON

Table 9 shows the comparisons between the existing methods in [9]–[14] and our proposal. We can see that our proposal overcomes the shortcomings in four aspects: anti peer compromise attack, anti MITM Attack, no additiaonal KDC required and encrypt the whole packets.

B. ID TRUSTED

In the proposal scheme, the entities' ID information is used as the input for public key generation and to be signed by the MSK so as to be trusted. This can make sure that there is no fake ID device involved in the secure communication.

Any unauthorized device with a fake ID can be identified immediately, since the digital signature verification will be invalid by its intended communicating counterpart.

TABLE 9. Security comparisons between existing method and our proposal.

	Shared the same masterkey [9] [10] [11]	ID-based scheme [13]	ECDH [12]	HMAC [14]	This proposal
Anti peer compromise attack	No	Yes	Yes	Yes	Yes
Anti MITM Attack	Yes	Yes	No	Yes	Yes
No additiaonal KDC required	Yes	No	Yes	Yes	Yes
Encrypt the whole packets	Yes	Yes	Yes	No	Yes

C. PERFORMANCE ANALYSIS

Next the performance based on evaluation from Klinc *et al.* [23] and Rebali *et al.* [24] will be analyzed. All of the off-line processes will not be calculated. The calculation process is broken down into steps evaluated in [23], they are: RNG (Random Number Generation), H (Hash), PM (Point Multiplication), PA (Point Addition), ENCB (Symmetric Encryption), DECB (Symmetric Decryption), all of the other XOR, multiplication and division will also be ignored, since they take too little time to impact the evaluation result. The main calculation time can be classified as expended by entity A and entity B.

Firstly, we calculate ECC signature verification time as: $1H + 2PM + 2PA$, denoted as ECCSIG.

Secondly, we calculate entity A's time cost:

$$1H + 1ECCSIG + 2RNG + 2PM + 1PA + DECB = 2H + 4PM + 3PA + 2RNG + DECB.$$

Similarly entity B's time cost is:

$$1H + 1ECCSIG + 1PM + 1PA + 1ENCB = 2H + 3PM + 3PA + 1ENCB.$$

According to the timing evaluation value in Table 10.

TABLE 10. Evaluation time.

Scheme	Time(ms)
RNG	0.539
H	0.0023
PM	2.226
PA	0.0288
ENCB	0.0046
DECB	0.0046

The evaluation time of both entities to complete the secure communication secret key establishing cryptography calculation is around $16.9ms$, which can be marked as t_{key}. The evaluation platform in referance [23] is a personal computer configured with Intel Pentium Dual CPU E2200 2.20GHz processor, 2048MB of RAM and Ubuntu 12.04.1LTS 32 bit operating system.

Besides, the extra data overload transmitting time between entity A and B during secure channel setup should be considered. The length of ID is 10 bytes(MAC + IP address), the length of Hash is 20 bytes(SHA1), the length of public key signature and the two public keys are 28 bytes each (ECC-NIST192), the RND is 16 bytes in plaintext or symmetric encryption mode(AES-256), and 56 bytes in our proposed asymmetric encryption mode. So the total data overhead is:

$$10 \times 2 + (28 + 28 + 28) \times 2 + 56 + 16 \times 2 = 276\ bytes.$$

Assume the data speed is 115200bps, the extra data transmission time is around: $19.2ms$, can be marked as t_{data}.

After the secure channel setup, the extra time expending is only the symmetric block encryption(AES-256), for a 256 bytes data package the extra time required for encryption or decryption is $(256/16) \times 0.0046 = 0.0736ms$, can be marked as t_{comm}.

For real system time cost the t_{key} and t_{data} should be estimated by adding up together, and t_{comm} can be estimated separately.

V. CONCLUSION AND FUTURE WORKS

In this paper the trusted-ID referenced key scheme for setup the secure SCADA communication channel with the help of a device sDTU in end-paired mode is proposed.

This proposal will solve the unsafe problem in end-to-end communication of SCADA systems in iron and steel plants or other applications with a practical and easy to implement solution.

The trusted-ID referenced idea will also make it easy for SCADA owners to clearly manage the assesses with a reliable identifier.

Though a relatively better approach for solving the SCADA secure communication problems is achieved, there are also some other points will be re-visited for the future study such as dynamically ID modification.

There will be another scenario for updating the ID information and related key pairs. And how to modify these sensitive information under authorization and access control also need to be further considered.

ACKNOWLEDGEMENT

The authors would like to express their sincerely thanks to all of whom providing excellent suggestions.

REFERENCES

[1] R. M. Lee, M. J. Assante, and T. Conway. *German Steel Mill Cyber Attack*. Accessed: Dec. 30, 2014. [Online]. Available: https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

[2] *Guide to Industrial Control Systems (ICS) Security*, NIST, Gaithersburg, MD, USA, 2015.

[3] C. Krushna and S. Magesh, "Analysis of vulnerabilities in the protocols used in SCADA systems," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 4, no. 3, pp. 1014–1019, Mar. 2015.

[4] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32910–32924, 2018.

[5] D. Beresford, *Exploiting Siemens Simatic S7 PLCs*. Accessed: Jul. 8, 2011. [Online]. Available: https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf

[6] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.

[7] A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi, and A. Y. Zomaya, "An efficient data-driven clustering technique to detect attacks in SCADA systems," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 893–906, May 2016.

[8] V. T. Alaparthy and S. D. Morgera, "A multi-level intrusion detection system for wireless sensor networks based on immune theory," *IEEE Access*, vol. 6, pp. 47364–47373, 2018.

[9] R. Dawson, C. Boyd, E. Dawson, J. Manuel, and G. Nieto, "SKMA: A key management architecture for SCADA systems," in *Proc. Australas. Inf. Secur. Workshop-Net Secur.* Hobart, TAS, Australia, 2006, pp. 183–192.

[10] S. Lee, D. Choi, C. Park, and S. Kim, "An efficient key scheme for secure SCADA communication," *Proc. World Acad. Sci., Eng. Technol.*, vol. 35, pp. 458–464, 2008.
[11] D. J. Kang, J. J. Lee, B. H. Kim, and D. Hur, "Proposal strategies of key management for data encryption in SCADA network of electric power systems," *Int. J. Elect. Power Energy Syst.*, vol. 33, pp. 1521–1526, Nov. 2011.
[12] A. Rezai, P. Keshavarzi, and Z. Moravej, "Secure SCADA communication by using a modified key management scheme," *ISA Trans.*, vol. 52, pp. 517–524, Jul. 2013.
[13] Y.-H. Lim, "IKMS—An ID-based key management architecture for SCADA system," in *Proc. IEEE Int. Conf. Netw. Comput.*, Gyeongsangbuk-do, South Korea, Sep. 2011, pp. 139–144.
[14] I. H. Lim *et al.*, "Security protocols against cyber attacks in the distribution automation system," *IEEE Trans. Power Del.*, vol. 25, no. 1, pp. 448–455, Jan. 2010.
[15] L. Cheng. *The Spear to Break the Security Wall of S7Commplus*. Accessed: Dec. 7, 2017. [Online]. Available: https://www.blackhat.com/docs/eu-17/materials/eu-17-Lei-The-Spear-To-Break%20-The-Security-Wall-Of-S7CommPlus.pdf
[16] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
[17] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO*. Santa Barbara, CA, USA, 1985, pp. 417–426.
[18] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*. New York, NY, USA, 1985, pp. 47–53.
[19] D. Boneh and M. Franklin, "Identity-based encryption from the weilpairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.
[20] *Information Technology–Security Techniques–Encryption Algorithms—Part 5: Identity-Based Ciphers*, Standard ISO/IEC 18033-5:2015, 2015.
[21] M. Abadi and M. R. Tuttle, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, pp. 18–36, 1990.
[22] G. Li, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, Oakland, CA, USA, May 1990, pp. 234–248.
[23] H. H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 2, pp. 1005–1023, 2nd Quart., 2014.
[24] Y. Rebahi, J. J. Pallares, N. T. Minh, S. Ehlert, G. Kovacs, and D. Sisalem, "Performance analysis of identity management in the session initiation protocol (SIP)," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, Doha, Qatar, Mar. /Apr. 2008, pp. 711–717.

JUNLEI QIAN was born in Tangshan, Hebei, China, in 1977. She received the B.S. degree in automation from Yanshan University, Qinhuangdao, in 2000, and the M.S. degree in control theory and control engineering from the University of Science and Technology Beijing, Beijing, China, in 2008. She is currently pursuing the Ph.D. degree with Yanshan University.

Since 2001, she has been a Lecturer with the North China University of Science and Technology. Since 2015, she has also been a Visiting Scholar with the University of Regina, Saskatchewan, Canada. Her research interests include process control systems and applications, and security of industrial control systems.

CHANGCHUN HUA received the Ph.D. degree in electrical engineering from Yanshan University, Qinhuangdao, China, in 2005. He 'was a Research Fellow with the National University of Singapore, Singapore, from 2006 to 2007. From 2007 to 2009, he was with Carleton University, Ottawa, ON, Canada, funded by the Province of Ontario Ministry of Research and Innovation Program. From 2009 to 2011, he was with the University of Duisburg-Essen, Duisburg, Germany, funded by the Alexander von Humboldt Foundation.

He is currently a Full Professor with Yanshan University. He has authored or coauthored more than 110 papers in mathematical, technical journals, and conferences. He has been involved in more than ten projects supported by the National Natural Science Foundation of China, the National Education Committee Foundation of China, and other important foundations. His current research interests include nonlinear control systems, control systems design over networks, teleoperation systems, and intelligent control.

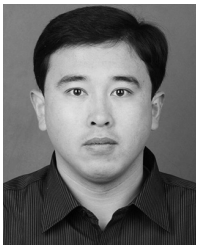
XINPING GUAN (SM'04–F'18) was a Professor and the Dean of Electrical Engineering, Yanshan University, China. He is currently a Chair Professor of Shanghai Jiao Tong University, China, where he is also the Deputy Director of University Research Management Office, and the Director of the Key Laboratory of Systems Control and Information Processing, Ministry of Education of China.

He has authored or coauthored four research monographs, more than 270 papers in IEEE Transactions and other peer-reviewed journals, and numerous conference papers. His current research interests include industrial cyber-physical systems, wireless networking and applications in smart city and smart factory, and underwater sensor networks. As a Principal Investigator, he has finished/been working on many national key projects. He is the Leader of the prestigious Innovative Research Team of the National Natural Science Foundation of China. He is an Executive Committee Member of the Chinese Automation Association Council and the Chinese Artificial Intelligence Association Council. He was elevated to IEEE Fellow, in 2017. He received the First Prize of Natural Science Award from the Ministry of Education of China, in 2006 and 2016, and the Second Prize of the National Natural Science Award of China, in 2008. He was a recipient of the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award, in 2008. He is a National Outstanding Youth honored by NSF of China, Changjiang Scholar by the Ministry of Education of China, and State-level Scholar of New Century Bai-Qian-Wan Talent Program of China.

TIEFENG XIN received the B.S. degree in precise instrument and the M.S. degree in precise instrument and mechanics from the Harbin Institute of Technology, Harbin, China, in 1989 and 1994, respectively.

Since 1996, he has been starting his career in smart card security field, and worked in several smart card companies as a Developing Manager. From 2005 to 2010, he was the Research and Development Director with Oburther Card System, Beijing, China. From 2010 to 2013, he was the Research and Development Director of the China Electronic Power Hwaray Technology Co., Beijing, China. His work experience is mainly in information security field, mainly focused on cryptography application. He developed key management system for bank card issuing, and embedded security module for smart power meter.

He is currently the Information Security Product Line Manager in China Electronic Power Hwaray Technology Co., is responsible for Industrial Control System security solution design and implementation.



LIMIN ZHANG was born in Hebei, China, in 1982. He received the M.S. degree in applied mathematics from the Hebei University of Science and Technology, in 2008, and the Ph.D. degree in control science and engineering from Yanshan University, Hebei, China, in 2016. He is currently with Hengshui University as an Associate Professor. His research interests include system identification, data mining, and machine learning.