Formal Security Analysis of OPC UA Protocol in Industrial Control System

Tao Feng, Zhuang-yu Ma, and Jun-li Fang

(Corresponding author: Zhuang-yu Ma)

School of Computer and Communication, Lanzhou University of Technology

Lanzhou,Gansu 730000,China

Email: 530901638@qq.com

(Received Aug. 30, 2021; Revised and Accepted Apr. 3, 2022; First Online Apr. 23, 2022)

Abstract

The OPC UA is a protocol used in the interaction between factory equipment. It is also widely employed in the industrial Internet industry, where long-distance data sharing and transmission between devices, multi-device interconnection, etc., must be realized. Therefore, it is essential to study the security of industrial control system protocol. However, most previous research on industrial control protocols mainly focuses on realizing the security of the protocol itself, lacking formal modeling and security evaluation, leading to some research gaps. Based on the previous colored Petri net theory, combined with the Dolev-Yao attacker model, this paper tries to make targeted improvements to analyze the security of the OPC UA protocol. First, based on the colored Petri net theory and CPN Tools, the security mechanism of the protocol is verified for consistency. Then the Dolev-Yao attacker model is introduced to evaluate the security of the original model of the protocol. By analyzing the security mechanism of the protocol, some issues, including the security of random numbers in OPC UA protocol and the deceptive attack of identity authentication attributes, have been found. Besides, some corresponding improvement projects are given. Finally, we used CPN Tools to verify the security of the new project. We also found that adding the recipient's public key to the message and the key packaging mechanism can effectively prevent attacks against the protocol, improving the protocol's security.

Keywords: Colored Petri Nets; Formal Analysis; Key Packaging; OPC UA Protocol; Security Evaluation

1 Introduction

With the gradual integration of informatization, industrial control systems are widely employed in many fields in a country and have become the focus of many countries to improve their comprehensive national strength. The OPC UA protocol are widely used, providing communicative connections for both mutually independent factory equipment and platforms.Since the Stuxnet attack occurred in 2010, the global industrial control system attack incidents have shown a spurt of growth [25], Industrial control security has already become an issue that cannot be ignored by government agencies. The security of industrial control systems is related to national strategic security,to a certain extent.Therefore, security has become the top priority of OPC UA protocol to expand the scope of use.

Industrial control systems differ from other systems.Due to the long lifespan of equipment and having difficulty in repairing vulnerabilities, the security must be carefully checked before deploying protocols and standards.Igure [11] and Patel [20] and others emphasized the lack of formal verification of traditional industrial protocols,and pointed out that formal verification is very important for evaluating the security of the protocol and discovering the loopholes of the protocol.Therefore,it is significant for both the formal analysis of OPC UA protocol and the security of OPC UA protocol.

The main contributions of this article include three aspects:

1) Adopt a detection method based on the previous colored Petri net theory and the Dolev-Yao attacker model;

2) Because of the extreme importance of two subprotocols of the OPC UA handshake, including Open Secure Channel and Create Session, hierarchical colored Petri nets (HCPN) for formal modeling, they will be employed to make analysis and verify the consistency of the sub-protocol models. Introduce the Delov-Yao attacker model to evaluate the security of the sub-protocols, and discover attacks against the sub-protocols;

3) As to the attack on the sub-protocol, the author proposes a security enhancement scheme, where the recipient's public key to the message is added, a key wrapper mechanism to prevent the identity of the random number from being deceived and forged is used and the security the new scheme is verified.

2 Related Work and Theories

Most of the researches on the security of industrial control protocols is limited to the written norms of human language and the realization of its own security, lacking the use of formal methods to be verified. The methods of formal analysis of the protocol mainly include modal logic, theorem proof, type checking and model checking. Although the modal logic method adopted in the literature [14] can verify whether the protocol meets its security goals through a series of reasoning, the logic is too abstract, leading to covering the state information of the protocol, the message exchange and aggressive behavior of the protocol cannot be described in more detail. Literature [22] analyzed the advantage of the theorem proving method, namely, being able to analyze the operation of infinite subjects participating in the protocol. Besides, the disadvantage of the method is that the process cannot be fully automated and requires manual intervention.

Literature [7] pointed out that the type detection method can verify the security properties of the protocol, but cannot find out the aggressive path. Literature [4] pointed out that the model checking method can verify whether the protocol meets the security properties, but cannot verify the correctness of the protocol and it also causes the problem of state space explosion. In terms of theoretical research, the author introduced the security mechanism of OPC UA protocol in parts 2, 4 and 6 of the technical document of OPC unified architecture [12,13,17]; Literature [10] deeply interprets the security issues of OPC UA and proposes security optimization strategies; Literature [21] uses Proverif, an encryption protocol verification tool, to check the security properties of the OPC UA protocol; Literature [6] introduced remote authentication and hardware authentication-based encryption to OPC UA, improving the security of OPC UA client-server communication; Literature [16] studied the security status of OPC UA,improving the communication efficiency by improving the encryption efficiency.

Compared with the previous researches, based on the colored Petri net theory and an improved Dolev-Yao attacker model, a detection method is proposed to analyze the security of the protocol, verify the consistency of the protocol model, then introduce the attacker model to evaluate the security of the protocol, propose a targeted security enhancement scheme for the evaluative results, and finally verify the security of the new scheme.

2.1 Overview of OPC UA Protocol

OPC UA, namely, OPC unified architecture, is a new industrial control protocol based on Web services, which was extended in the basis of traditional OPC foundation. Besides, it has been widely employed in long-distance interaction between factory equipment and multi-device interconnection. It also has the functions of cross-platform, integrated address space, encapsulation of general service interfaces, and definition of security models [9,18]. What differs from the traditional OPC protocol is that the OPC UA protocol can realize the long-distance data exchange of various industrial systems and equipment through the Internet, realize the communication between the client and the server and meet the needs of data exchange at all levels of the industrial control system.

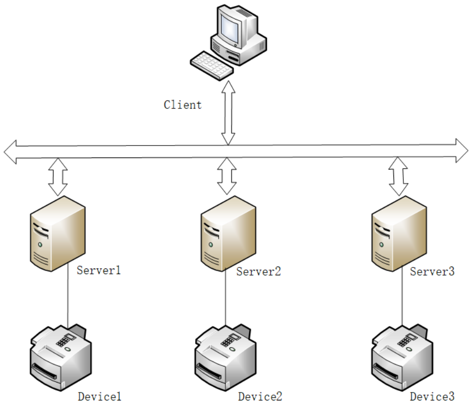

Figure 1: OPC UA communication mode

The communication mode of OPC UA agreement is shown in Figure 1 The client makes a service request, and the server receives the request from the client, performing a series of operations in the address space, and giving a response through the API. It can be seen that the communication method of OPC UA is closer to the modern communication method, which is more conducive to system management and application development.

2.2 Introduction to OPC UA Security Mechanism

Industrial Control System (ICS) not only needs to realize the real-time transmission of control data, but also ensure the safety of data during the transmission process. Therefore, in the application of OPC UA in the field of industrial control, security is of the extreme importance. OPC UA defines a security model in order to ensure the communication security of OPC UA, as is shown in Figure 2.

The OPC UA security model can be divided into three layers. The bottom is the transport layer, which is the basis for ensuring security communication, and is also one that is easier to be attacked, such as a denial of service attack; What is a bit higher than the transport layer is the communication layer where the secure channel can be established. The communication layer adopts asymmetric encryption, digital signature and other technologies to ensure the confidentiality and integrity of the channel. At the same time, identity authentication and authorization mechanisms are used to ensure the authenticity of communications; The top one is the application layer. Based on the secure channel, the client and server make a communication at the application layer through negotiation, which is used to transmit real-time data and operate in-

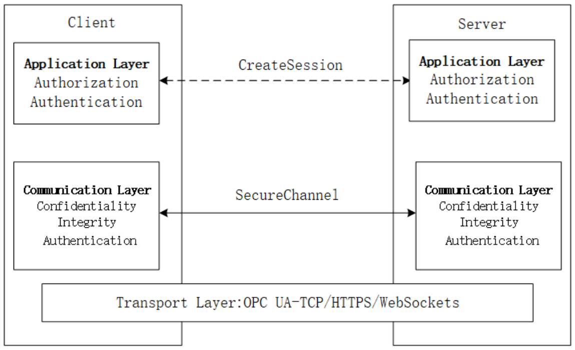

Figure 2: OPC UA security model

structions between devices. Making a communication can verify and authorize the client's identity.

This security model provides a certain foundation for the security of OPC UA, and provides guidance for analyzing the security of the two handshake sub-protocols of OPC UA.

2.3 Colored Petri nets and Modeling Tools

German scientist Carl Adam Petri first proposed the concept of Petri net and then many extensive concepts appeared, such as time Petri net, stochastic Petri net, CPN, etc. [1, 15, 24] Colored Petri Nets (Colored Petri Net, CPN) is an advanced form of traditional Petri Nets. Compared with Petri Nets which has no concept of types and modules, CPN has various advantages, including a variety of types, complicated operative data, rich and flexible color sets, definitive types of the function and the ability to describe hierarchical structures. Besides, it can provide the operative interface. These can be conducive to giving standardized definition of message flow models, which can be used to describe and analyze communication, distribution, and Protocols and systems with features such as synchronization and concurrency.

CPN Tools is a system modeling analysis tool developed by the Aarhus University team in Denmark. The model description language is composed of Petri net diagrams and CPN ML programming language. It uses good man-machine interface technology to perform graphical user interface (GUI) design. It also has functions of incremental syntax checking and code generation. It can be edited, simulated, used to analyze the state of space and other features of the model and can accurately locate errors in the model through the feedback mechanism, which ensures the correctness of the model to a certain extent. In addition, it also supports for time CPN and hierarchical CPN. With the help of CPN Tools, users can not only model easily, but also simulate and analyze parallel systems [2,3].

3 OPC UA Sub-protocol Modeling

Faced with modeling large-scale and more complex protocols with Petri nets, it is more complicated to adopt the traditional CPN model. Therefore, at this time we need to adopt the idea of modular programming, namely, the concepts of substitution transitions and port places in CPN Tools. The network or protocol structure is divided into multiple modules and the network with substitution changes is a multi-level network. We can first establish a simplified network top-level model more broadly and then further refine the sub-pages through the substitution transitions in the top-level model. The layered modeling method can reduce model complexity and improve reusability. This paper divides the OpenSecureChannel and CreateSession sub-protocol modeling into three levels:top level, middle level and bottom level.

3.1 OpenSecureChannel Sub-protocol Message Flow Model

The OpenSecureChannel sub-protocol aims to achieve identity verification between the client and the server by exchanging two secret random numbers and derive a shared key for future communication. In addition, OPC UA has three security modes, including "None", "Signature" and "Signcryption" [19]. "None" shows that the secure channel is in an insecure state and the sub-protocol does not provide any security, but for compatibility. "Signature" means that the private key h(m)sk(x) associated with the OPC UA client certificate can be used for digital signature and the recipient can verify whether the transmitted message has been tampered with by a third party or not. The "signcryption" mode means that the transmitted message is not only signed, but also encrypted by the public key associated with the client certificate. Encryption is used to provide confidentiality for communication and signatures are used to provide authentication and integrity.

The message flow model of this sub-protocol is shown in Figure 3.

Step 1. The client needs to send a Get EndPoints request that asks information about the server;

Step 2. Discovery EndPoint uses the server's public key pk(S), security mode, security policy SP, and user policy UP to make a response, where both SP and UP are used to encrypt primitive negotiation;

Step 3. The client sends a request to OpenSecureChannel the client's public key pk(C), and a random number NC to the server, using the server's public key pk(S) to encrypt and using the client's private key sk(C) to sign;

Step 4. The server sends OSCRes, random numbers N_S, ST, TTL as a response, and uses pk(C) encryption

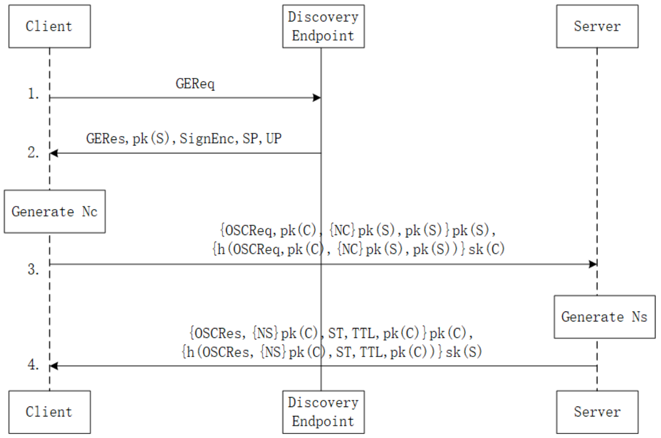
Client
Discovery Endpoint
Server
1.
GEReq
2.
GERes, pk(S), SignEnc, SP, UP
Generate Nc
{OSCReq, pk(C), {NC}pk(S), pk(S)}pk(S),
{h(OSCReq, pk(C), {NC}pk(S), pk(S))}sk(C)
3.
Generate Ns
{OSCRes, {NS}pk(C), ST, TTL, pk(C)}pk(C),
{h(OSCRes, {NS}pk(C), ST, TTL, pk(C))}sk(S)
4.
Client
Discovery Endpoint
Server

Figure 3: OpenSecureChannel sub-protocol message flow model

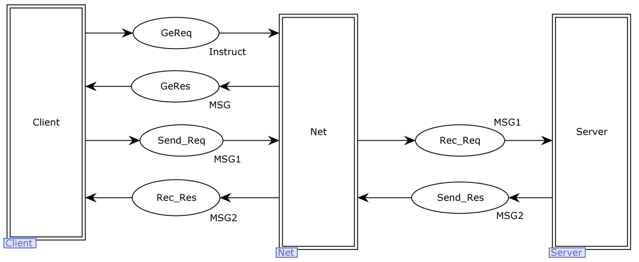

Figure 4: OpenSecureChannel sub-protocol top-level model

and sk(S) signature, where OSCRes stands for the response of the open secure channel, ST stands for the channel identifier, and TTL stands for its life cycle.

3.2 Establishment of OpenSecureChannel Sub-protocol CPN Model

This article is based on the OpenSecureChannl sub-protocol signcryption model of the message flow model for specific modeling.

The top-level CPN model of the OpenSecureChannel sub-protocol is shown in Figure 4, where the interaction process of the sub-protocol is abstractly described on the whole, including the client, server, communication network and transmitted messages of the protocol. In the top-level model of Figure 4, the double-line rectangle stands for the substitution transition and the ellipse stands for the message repository. The substitution transition Client on the left stands for the communication client, the substitution transition Net in the middle stands for the communication network and the substitution transition Server on the right stands for the server.

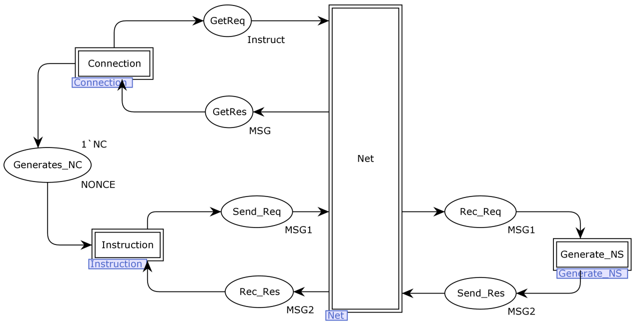

Figure 5: The middle layer model of the OpenSecureChannel sub-protocol

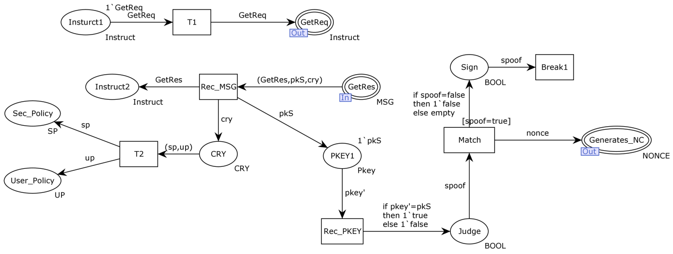

Figure 6: The internal model of the substitution transition Connection

The middle model of this sub-protocol consists of 4 substitution transitions and 7 places which are shown in Figure 5. The process of achieving connection information between the client and the server is represented by the substitution transition Connection; The process in which the client sends an OpenSecureChannel request instruction to the server is represented by the substitution transition Instruction; The process of generating random numbers from the server to the client is represented by the substitution transition Generate_NS.

The bottom model of this sub-protocol includes 4 parts. According to the order of interaction between the client and the server, the connection establishment, request instruction and random number generation will be introduced as fully as possible. The internal model of the substitution transition Connection is shown in Figure 6. Transition T1 is used to make a request for information from the server and the response message from the client is received by the place GetRes through the transition Rec MSG processing, the received server's public key pk(S) is received through the transition Rec_PKEY. It is verified by Transition Match. After the verification is correct, the storehouse GenerateNC generates a random number N_C. If the verification fails, the operation will be ended.

Figure 7 depicts the internal model of substitution transition Net. The transition DisEndPoint simulates the transmission path where the client sends a request to the server to obtain terminal description information, the transition OSCReq and OSCRes, respectively, simulate the transmission path of the client initiating the OpenSecureChannel makes a request for information to the server

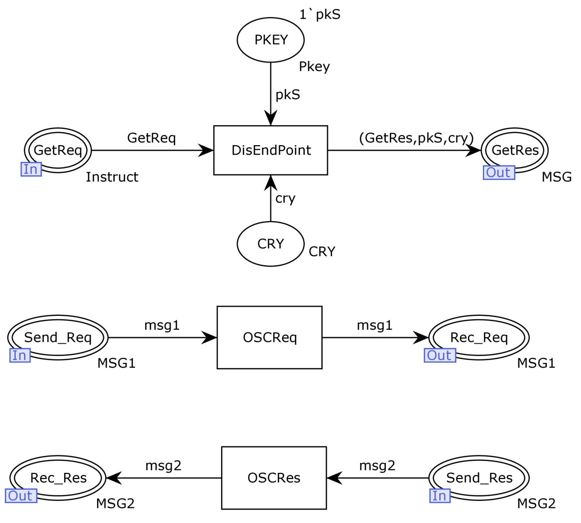

Figure 7: The internal model of the substitution transition Net

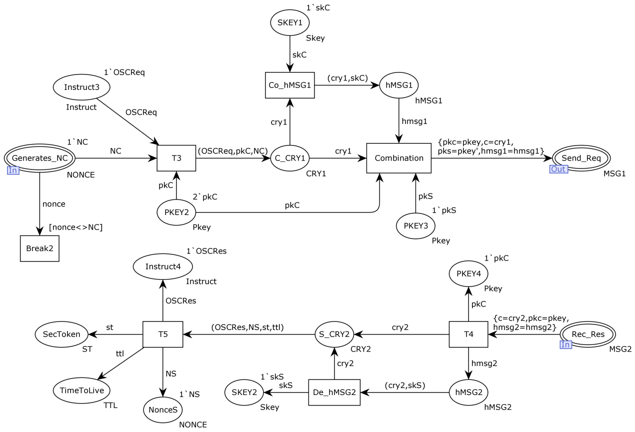

Figure 8: The internal model of the substitution transition Instruction

and the transmission path of the server in response to the request information to the client.

Figure 8 depicts the internal model of the Alternative Transition Instruction. Random number N_C is generated by the repository Generates_NC, transition T3 integrates the random number N_C that has been received, client public key pk(C) and request OSCReq. Transition Combination combines the signed information and encrypted information. Finally, this information is sent to the server through the repository Send_Req; transition Break2 means that the random number N_C has been received, verification failed to perform the termination operation, The place Rec_Res is used to receive the data information sent by the server. Transitions T4 and T5 decompose the received safety data.

Figure 9 depicts the internal model of the substitution transition Generate_NS. The place Rec_Req is employed to receive the data information sent by the client to the server, transition Req_Message decomposes the security data which have been received and transition Rec_S' verifies the random number N_C which have been received. After the verification is finished, the random number N_S will be generated by the place Gen_NS, If the verification fails, operation Break3 will be ended; Transition T7 integrates the response command OSCRes which have been received, identifiers and its life cycle; transition Combination' combines all the information which have been received; Finally, this information is sent to the client through the place Send_Res.

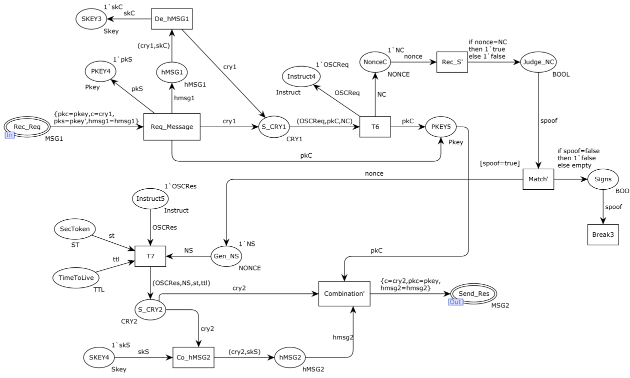

Figure 9: The internal model of the substitution transition Generate_NS

3.3 OpenSecureChannel Sub-protocol Model Consistency Verification

The analysis tool of state space in CPN Tools will be employed to verify the consistency of the original CPN model of the sub-protocol. We first give the expected results of the original model which was established. The model will successfully perform request, make a response and verify random number where operations are never ended in the process of the entire interaction.

Table 1 shows the results of state space in the original model of the sub-protocol. It can be clearly found that the number of state space nodes, directed arcs and strongly connected nodes and strongly connected arcs is equal, which shows that all state nodes of the original model that were established are reachable. In addition, there are no infinite loops and iterative behaviors in the state space. The number of main state nodes and active transitions are both 0, indicating that there is no reachable state in the original model, and there is no active transition in the active state; The existence of a dead node indicates that any transition under this node are not enabled; There are 3 dead transitions Break1, Break2 and Break3. the transition Break1 means that after the establishment of the connection the operation will be ended once it fails to verify pk(S).The transition Break2 shows the operation is ended because of the error of random number N_C, the transition Break3 stands for the ended operation that generates the failure of ver-

Table 1: State space analysis of the original model of OPC UA OpenSecureChannel sub-protocol

Categorys	Numbers	Name
State space node	30	/
Directed arc	59	/
Strongly connected node	30	/
Strongly connected arc	59	/
Master state node	0	/
Dead node	1	[30]
Death transition	3	Break1/Break2 /Break3
Live transition	0	/

ification of random number N_S. The existence of these three dead transitions reveals there is no failure of operation and verification for the original model and it accords with the expected results.

3.4 CreateSession Sub-protocol Message Flow Model

The clients are allowed to send credentials (username and password) through the secure channel established in the CreateSession sub-protocol.This sub-protocol follows the security model of sub-protocol that is used in the OpenSecureChannel and uses a derived symmetric key. Therefore, the signature will depend on the message authentication code (MAC) once symmetric encryption is used. The message flow model of this sub-protocol is shown in Figure 10. The message sent by the client is encrypted by KSC and signed by KSigcs, and the message sent by the server is encrypted by KSC and signed by KSigsc. Step 1: The client sends a CreateSession request, a random number N_C and the client's public key pk(C) to the server. After encrypting with KCS and signing by KSigcs, the client sends them to the server; Step 2: the server sends SigNc=pk(C),$N_C$sk(S), CreateSession response, random number N_S, server public key pk(S) through signcryption processing and sends it to the client as a response to make a conversation; Step 3: the client uses the signature SigNs of the random number N_S to send pk(C), ActivateSession request and user credentials through signcryption processing, and sends them to the server as a request to activate the conversation; Step 4: the server makes a response for ActivateSession and a fresh random number N_S2, which also undergoes sign cryption processing, as the client, changes the interrogate of the conversation and makes a response when the conversation exceeds the time limit.

3.5 Estabishment of CPN Model of CreateSession Sub-protocol

In this paper, the specific modeling is based on the message flow model of the signcryption mode of the CreateSession sub-protocol.The top-level CPN model of the CreateSession sub-protocol is shown in Figure 11, where the interaction process of the sub-protocol is abstractly described on the whole, including the client, server, communication network and transmitted messages of the protocol. The substitution transitions Client, Net, Server stand for client, communication network and server respectively.

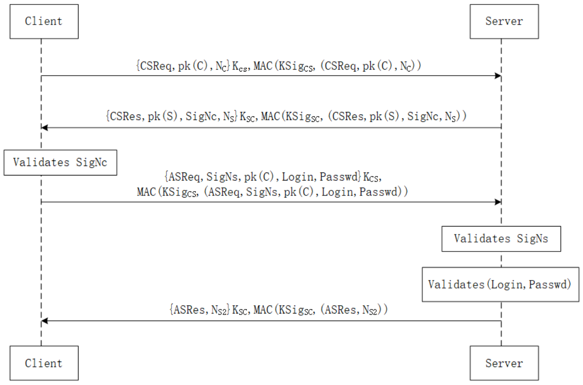
Client
Server
$\{CSReq, pk(C), N_C\}K_{cs}$, $MAC(KSig_{CS}, (CSReq, pk(C), N_C))$
$\{CSRes, pk(S), SigNc, N_S\}K_{SC}$, $MAC(KSig_{SC}, (CSRes, pk(S), SigNc, N_S))$
Validates SigNc
$\{ASReq, SigNs, pk(C), Login, Passwd\}K_{CS}$,
$MAC(KSig_{CS}, (ASReq, SigNs, pk(C), Login, Passwd))$
Validates SigNs
Validates(Login, Passwd)
$\{ASRes, N_{S2}\}K_{SC}$, $MAC(KSig_{SC}, (ASRes, N_{S2}))$
Client
Server

Figure 10: CreateSession sub-protocol signcryption mode message flow model

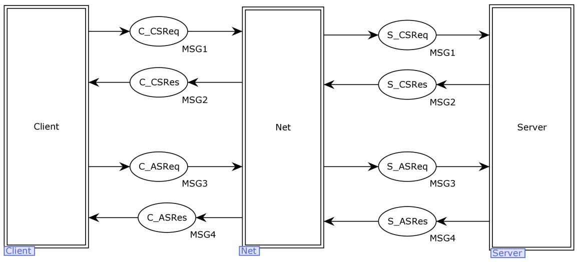

Figure 11: Top-level model of CreateSession sub-protocol

The middle model of this sub-protocol consists of 5 substitution transitions and 9 places which are shown in Figure 12. The process of achieving connection information between the client and the server is represented by the substitution transition CreateSession and CreateSession;The process of activating the session from the client to the server is represented by the substitution transitions ActivateSession and ActivateSession';The Net stands for communication network.

The bottom model of this sub-protocol includes 5 parts. The internal model of the substitution transition CreateSession is shown in Figure 13. Transition T1 integrates the received CSReq, pk(C), and N_C, and transition Combination1 merges all messages which has been received. Finally, it is sent to the server through the place S_CSReq; The place R_CSRes is used to receive the security information sent by the server and the transitions C_CSRes,T2 and T3 are used to decompose the security information which has been received; The transition Re_NC is used to verify the random number NC which has been received. If the verification succeeds,the

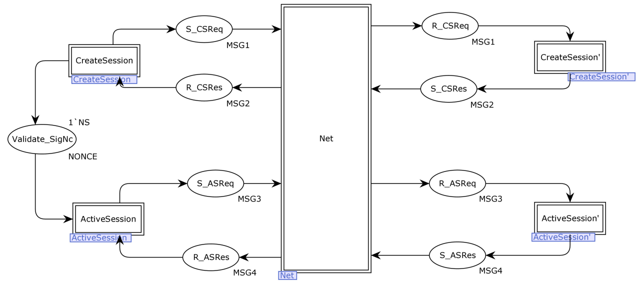

Figure 12: The middle layer model of CreateSession sub-protocol

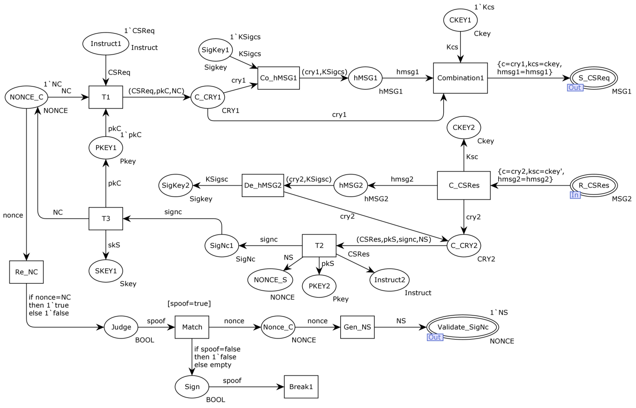

Figure 13: The internal model of substitution transition CreateSession

Validate_SigNc will generate a random number NS. But if it fails, the operation will be ended.

Figure 14 depicts the internal model of substitution transition Net.The transitions CreateSessionReq and CreateSessionRes imitate the transmission path of the client and server to create the requests for session and response, the transitions ActivateSessionReq and ActivateSessionRes imitate the transmission path of the client and server to activate the session request and response activation respectively.

Figure 15 depicts the internal model of the substitution transition CreateSession'. The place R_CSReq is used to receive the message sent by the client and then it is decomposed by the transition Rec_CSReq, T6, and T7; Transition T8 integrates the random number N_S which has been generated, request for response CSRes, SigNc, and server public key pk(S); Transition Combination1' combines the received information and sends it to the client through the place S_CSRes.

Figure 16 depicts the internal model of the substitution transition ActivateSession. The place Validate_SigNc sends the random number NS that are generated to the transition C_SigNs and then integrates the received keys pk(S) and sk(C).Transition T4 integrates the received activation request ASReq, user credentials login, pwd, and key pk(C), Combination2 will merge all received messages and send them to the server through the place S_ASReq; The place R_ASRes is used to receive the response message sent by the server and it is decomposed by the transition S_ASRes and T5.

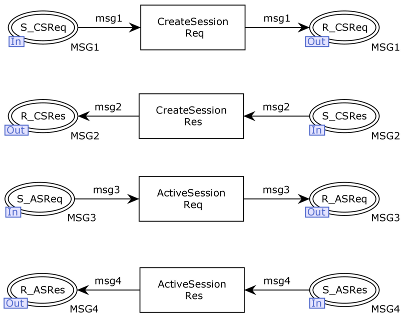

Figure 14: The internal model of substitution transition Net

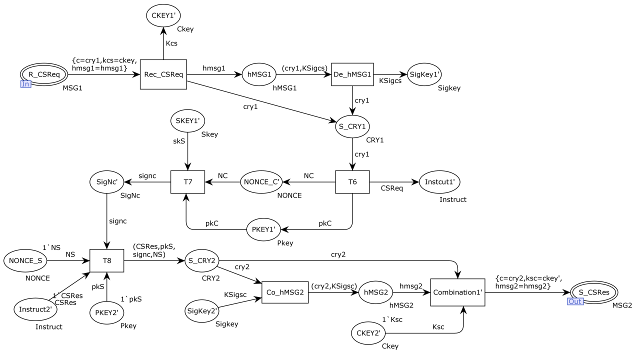

Figure 15: The internal model of substitution transition CreateSession'

Figure 17 depicts the internal model of the substitution transition ActivateSession'. The repository R_ASReq is used to receive the request information sent by the client and decompose it through the transitions S_ASReq, T9, and S_SigNs, the places Vali_User and Vali_SigNs are used to verify user credentials (assuming the user name is "admin" and the password is "123456") and the random number N_S, respectively, If it is successfully verified, the random number N_S2 is generated through the transition Match_User. But if not, the operation Break2 is ended; The transition T10 integrates the received random number N_S2 and the activation response ASRes, merges them by the transition Combination2' and sends them to the client through the place S_ASRes.

3.6 Consistency Verification of CreateSession Sub-protocol Model

First of all, the authors give the expected results of the original model established. The model will successfully perform the request and make a response for session creation and activation and there is no interruption of operation in the process of the entire interaction. Table 2

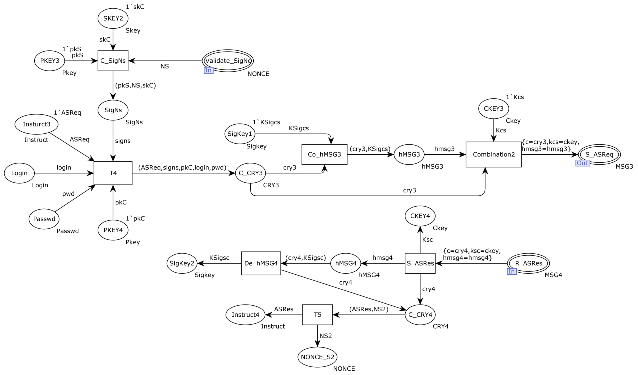

Figure 16: The internal model of substitution transition ActivateSession

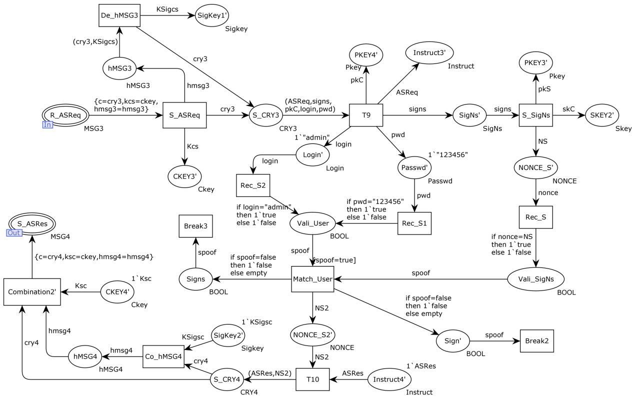

Figure 17: The internal model of substitution transition ActivateSession'

shows the results of state space in the original model of the sub-protocol, which is totally similar to Table 3. The number of state space nodes, directed arcs and strongly connected nodes, and strongly connected arcs is equal, All state nodes of the original model that are established are reachable, and there are no infinite loops and iterative behaviors in the state space; The number of main state nodes and the number of active transitions are both zero, indicating that there is no reachable state in the original model, and there is no active transition in the active state; The existence of a dead node indicates that any transition under this node are not enabled; There are 3 dead transitions Break1, Break2 and Break3.The transition Break1 means that the operation will be ended once the random number N_C verification fails during the creation of the session. The transitions Break2 and Break3 respectively represent the interruption of operation of the user credential and the random number N_S verification fails in the activation session, the existence of these three dead transitions show that the original model does not fail to pass the verification, which accords with the expected results.

Table 2: State space analysis of the original model of the OPC UA CreateSession sub-protocol

Categorys	Numbers	Name
State space node	64	/
Directed arc	168	/
Strongly connected node	64	/
Strongly connected arc	168	/
Master state node	0	/
Dead node	1	[64]
Death transition	3	Break1/Break2 /Break3
Live transition	0	/

4 Based on the Attacker's Security Assessment Model

Dolev and Yao published an important paper, which have a profound impact on the development of protocol security research [8]. The main contribution of this paper is to only analyze the security properties of the protocol itself based on the assumption that the cryptographic system is "perfect". At the same time, an attacker model with powerful computing power has been proposed, which can not only eavesdrop, intercept, tamper, and replay the messages interacted during the operation of the protocol, but also encrypt, decrypt, split and combine the information [5, 23]. In this way, we can concentrate on studying the inherent vulnerability and security of the protocol without caring about the security of the cryptographic algorithm.

Because the sub-protocol of OPC UA has a high degree of real-time performance and data frames are transmitted between the client and the server, this article attempts to add an attacker model to the network channel between the client and the server.

4.1 Introducing an Attacker's Security Assessment Model

According to the Dolev-Yao attacker model, the attacker has the powerful ability to initiate various man-in-the-middle attacks on network channels, Man-in-the-middle attacks of replay, spoofing and tampering are introduced to the network transport layer of the two sub-protocols of the OPC UA handshake.

As shown in Figures 18 and 19, man-in-the-middle attacks are added to the OpenSecureChannel and CreateSession sub-protocol network transport layers, including replay, tampering, and spoofing. Different color sets have different places and transitions functions. The red part of the places and transitions in the figure imitates a tampering attack, and the attacker attack and attack are introduced into the expression; The blue part of places and transitions in the figure imitates a replay attack. The transition TA intercepts the message of the initial operation of the sub-protocol that the attacker is going to split and is stored in the place DB, places CB1, CB2, CB3 store atomic information. The attacker's decomposition principle is adopted to decompose transition TC message; The message after the transition TD synthesizes the atomic

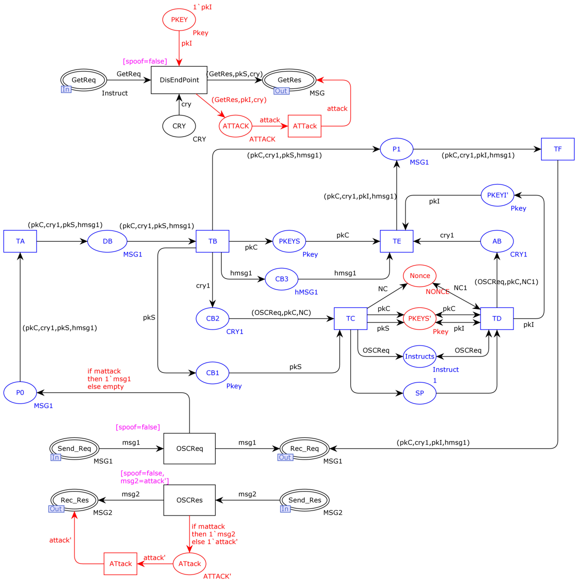

Figure 18: Security assessment model for the attacker of the OpenSecureChannel sub-protocol

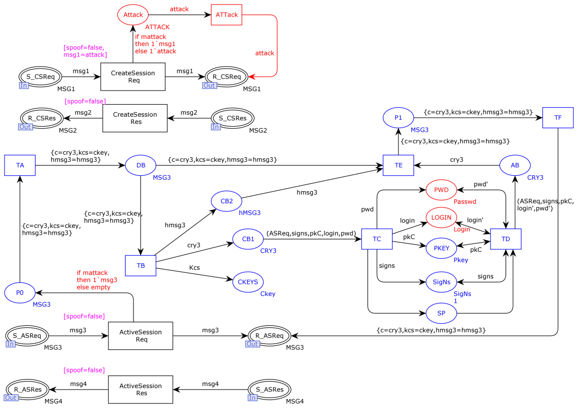

Figure 19: Security assessment model for the attacker of the CreateSession sub-protocol

message is stored in the place AB and concurrency control place SP in the process of synthesis should be introduce to restrict and limit; The transition TF is used to synthesize the last attack message which is sent to the port library. The purple part imitates a spoofing attack, covering all transitions in the process of network transmission in the two sub-protocols.

4.2 Security Evaluation of OPC UA Sub-protocol Model

Table 3 and Table 4 are the state space reports of the two sub-protocol attacker security assessment models. It can be seen that the number of state space nodes, directed arcs and strongly connected nodes and strongly connected arcs is same, which shows that all state nodes in the attacker model are reachable. Compared with the original model, the number of state space nodes and arcs does hardly increase after the attacker model is added, indicating that there will be no state space explosion after the attacker model is introduced, declining the size of the state space node and reducing the message that is not recognized by the receiver.

Table 3: State space comparison of OpenSecureChannel sub-protocol model

Categorys	Original	Attacker
State space node	30	184
Directed arc	59	402
Strongly connected node	30	184
Strongly connected arc	59	402
Dead node	1	4
Death transition	3	7

Table 4: State space comparison of CreateSession sub-protocol model

Categorys	Original	Attacker
State space node	64	87
Directed arc	168	213
Strongly connected node	64	87
Strongly connected arc	168	213
Dead node	1	1
Death transition	3	3

After comparison of dead transition between original model and attacker modelin Tables 3 and 4, it can be found that for the dead marking of the OpenSecureChannel sub-protocol become 4 and the dead transitions become 7.Through further inquiry and analysis, there are 3 dead markings and 4 dead transitions because of the introduction of replay and spoofing attacks. The intruder tampered with the destination of the data stream, bypassed the protection of replay attacks and generated an authentication attack, protocol produced unpredictable end state. For the CreateSession sub-protocol after adding the attacker model, the number of dead markings and dead transitions has not been changed and the attacker cannot obtain any credentials of the sub-protocol. It shows that the CreateSession sub-protocol meets the security attribute goals.

The security of the two sub-protocols of OPC UA needs to be evaluated by introducing an attacker model. Since the attacker's public key pk(I) is used in message 2 to tamper with the client's public key pk(C) and sent to the client, the destination of the message flow is changed, thus leading to the client to initiate a conversation with the attacker, After receiving it, the attacker uses his own private key to decrypt it and initiates a session with the server, thereby generating a spoofing attack on the client's identity verification by the random number N_C; In addition, random numbers are exchanged in the form of plain text

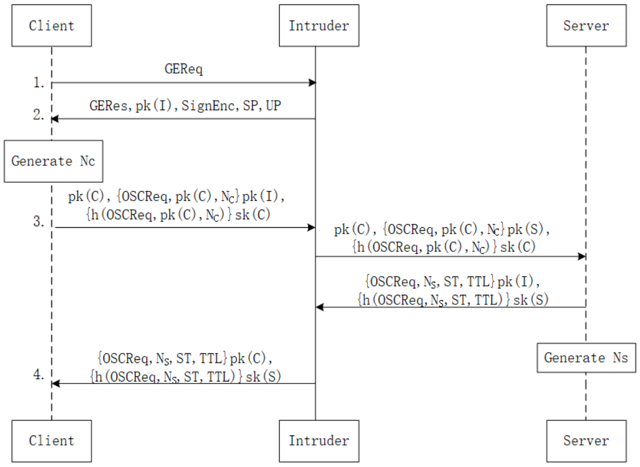
Client
Intruder
Server
1.
GEReq
2.
GERes, pk(I), SignEnc, SP, UP
Generate Nc
3.
pk(C), {OSCReq, pk(C), N_C} pk(I),
{h(OSCReq, pk(C), N_C)} sk(C)
pk(C), {OSCReq, pk(C), N_C} pk(S),
{h(OSCReq, pk(C), N_C)} sk(C)
{OSCReq, N_S, ST, TTL} pk(I),
{h(OSCReq, N_S, ST, TTL)} sk(S)
Generate Ns
4.
{OSCReq, N_S, ST, TTL} pk(C),
{h(OSCReq, N_S, ST, TTL)} sk(S)
Client
Intruder
Server

Figure 20: Attacks on N_C and N_S authentication

in the process of message flow interaction, which causes the confidentiality of random numbers.

Therefore, the OpenSecureChannel sub-protocol has spoofing attacks on the client and server authentication by the random number N_C and N_S respectively, and the confidentiality of the random number. As shown in Figure 20, Spoofing attacks that use N_C and N_S to authenticate the client and server respectively, Because the OPC UA protocol standard does not require the identity of the message recipient to be displayed, this attack is possible. Therefore, it allows the intruder to send the client's signed message to the server. This attack is similar to the man-in-the-middle attack of the NS (Needham-Schroeder) protocol.

5 New Scheme of OPC UA Protocol

5.1 New Plan Reinforcement Method

As for the security evaluation results, the authors in this paper add the recipient's public key to the message in the sub-protocol session in order to solve the spoofing attack of identity authentication, thus leading to preventing the intruder from resending the signed message to the tampering host. As for the confidentiality of random numbers, the key wrapping mechanism is used to replace all random numbers NC in message 3 with $N_C$pk(S), and replace all random numbers N_S in message 4 with $N_S$pk(C). Since the industrial control protocol has higher requirements for real-time performance, this scheme of security enhancement that improves the protocol on the message stream is adopted so that the real-time performance of the protocol will not be affected. Figure 21 shows message flow model that has been improved.

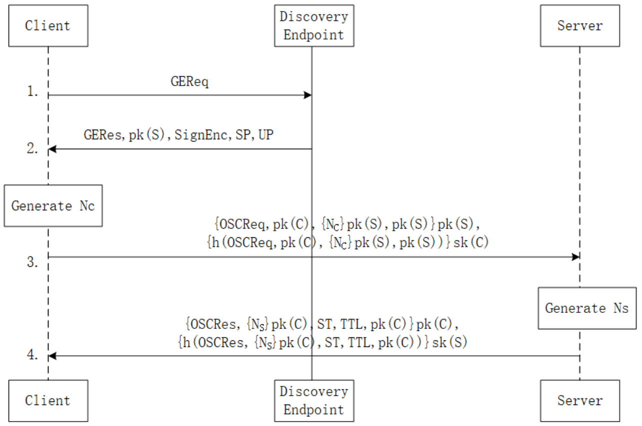
Client
Discovery Endpoint
Server
1.
GEReq
2.
GERes, pk(S), SignEnc, SP, UP
Generate Nc
3.
{OSCReq, pk(C), {N_C} pk(S), pk(S)} pk(S),
{h(OSCReq, pk(C), {N_C} pk(S), pk(S))} sk(C)
Generate Ns
4.
{OSCRes, {N_S} pk(C), ST, TTL, pk(C)} pk(C),
{h(OSCRes, {N_S} pk(C), ST, TTL, pk(C))} sk(S)
Client
Discovery Endpoint
Server

Figure 21: The message flow model of the new and improved OpenSecureChannel

5.2 OpenSecureChannel New Improvement Scheme Model

CPN modeling for the security enhancement scheme of the sub-protocol is performed. On the basis of the original model, the color sets Sec1 and Sec2 with the key packaging are newly added. The message flow in the sub-protocol has been changed, the top-level and middle-level models of the sub-protocol remain unchanged. This refers to the internal model that gives the substitution transitions, which have been changed.

Figure 22 shows the internal model of the new scheme of substitution transitions Instruction.The newly added places Sec1 and Sec2 are used to store the security data wrapped by the key, the random number N_C is generated by the place Generates_NC and the transition T4 will package the random number N_C that has been received and public key pk(S), and then integrated by Transition T5; Transition Combination combines the signed information and encrypted information. Finally, this information is sent to the server through the place Send_Req; Transition Break 2 means that verification of the random number N_C that has been received failed to perform the operation; The place Rec_Res is used to receive the data information sent by the server and the transition Reception, T6 and T7 indicate that the security data that has been received will be decomposed.

Figure 23 shows the internal model of the new scheme substitution transitions Generate_NS. The newly added places Sec1'and Sec2'store the security data wrapped by the key, the place Rec_Req is used to receive the data information sent by the client to the server, transition Reception',T8, T9 are used to decompose security data, transition Rec_S' verifies that the random number N_C that has been received generates a random number N_S after the verification is passed and executes the interruption of Break 3 operation if the verification fails; Transition T10 will package the random number N_S that has been received and public key pk(C) and integrate

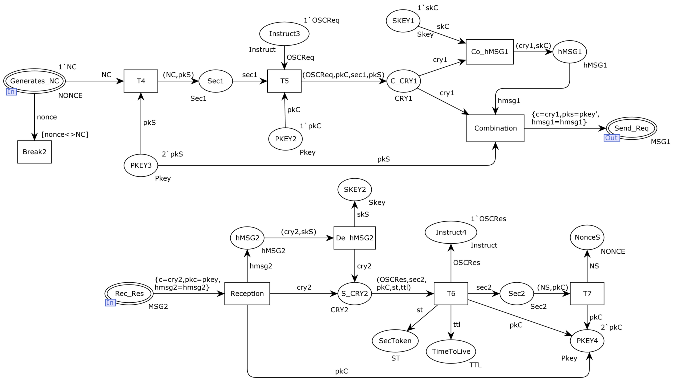

Figure 22: The internal model of the new improvement plan substitution transition Instruction

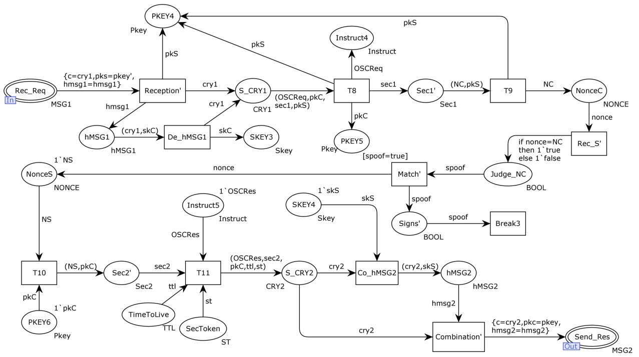

Figure 23: The new improvement plan substitution transition Generate_NS internal model

the response command OSCRes, identifier and its life cycle through the transition T11; Transition Combination'combines the signed information and encrypted information. Finally, the information is sent to the client through place Send_Res.

5.3 Safety Assessment Model of the New Scheme

Same as 4.1, we introduce the Dolev-Yao attacker model to the new scheme, and add man-in-the-middle attacks such as tampering, deception, and replay to the network layer of the new scheme of the OpenSecureChannel sub-protocol. As shown in Figure 24, the blue, red, and purple parts respectively simulate replay, tampering, and spoofing attacks.

5.4 Security Assessment of the New Scheme of the Sub-protocol

Table 5 shows the comparison of results of the state space in the OpenSecureChannel sub-protocol security evaluation model after the improvement. Because the wrapper mechanism of the key and the definition of related color

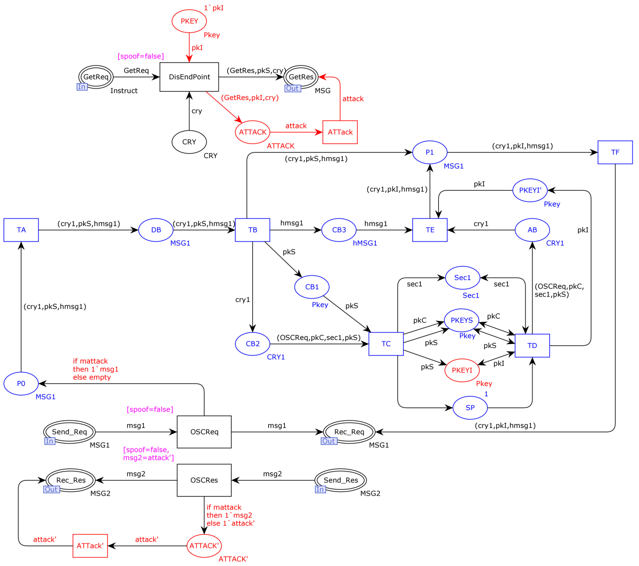

Figure 24: OpenSecureChannel new solution security evaluation model

sets are added to the message flow, the number of transitions and places is correspondingly increased and the number of state space nodes and directed arcs has also increased after improvement.

Table 5: State space comparison of OpenSecureChannel sub-protocol before and after improvement

Categorys	Before Improvment	New Scheme
State space node	184	264
Directed arc	402	536
Strongly connected node	184	264
Strongly connected arc	402	536
Dead marking	4	1
Death transition	7	6

It can be found after analysis that the number of dead marking is reduced by one,which is consistent with the number of dead marking in the original model of the sub-protocol.

This dead marking shows the final state of the protocol that has been performed after SML sentence analysis, indicating that there is not any attack on the new scheme. The number of dead transitions is reduced by 6 and the analysis shows that 3 dead transitions occurred during the operation of the protocol, which resulted in an error termination operation. Besides, the reason why the other three dead transitions are at the network level is that the attacker cannot set off an effective attack.

The attacker cannot obtain the receiver's public key and the confidentiality of the random number that is sent must be guaranteed, forcing the attacker not to initiate a spoofing attack, which reveals that the new scheme can protect against attacks from sub-protocol identity authentication and enhance the security of the protocol.

6 Conclusion

Colored Petri nets and Delov-Yao attack methods is adopted as the theoretical basis in this study and the OPC UA protocol between factory equipment is regarded as the research object, Because the two sub-protocols of OPC UA handshake, including OpenSecureChannel and CreateSession represent the core of the OPC UA protocol security, CPN Tools are employed to make formal modeling and do security assessment for these two sub-protocols. It has been found after the modeling and analysis of these two sub-protocols that a security enhancement scheme adding the recipient's public key to the message and the key packaging mechanism is proposed and CPN Tools is used to verify the security scheme.This study only set off attack to a man-in-the-middle in the protocol, analyzed the security of the protocol itself, but did not take other forms of attacks into consideration. The next research should consider whether the protocol has other security issues at other levels in the future study and can make analysis security in other forms of attacks.

Acknowledgments

This research is supported by The National Natural Science Foundation of China (Grant No. 61762060), Educational Commission of Gansu Province, China (Grant No.2017C-05), Foundation for the Key Research and Development Program of Gansu Province, China (Grant No.20YF3GA016). Tao Feng is the corresponding author.

References

[1] M. Abbaszadeh, S. Saeedvand, "Weak consistency model in distributed systems using hierarchical colored petri net," *Journal of Computers*, vol. 13, no. 2, pp, 236-243, 2018.

[2] D. Arena, F. Criscione, N. Trapani, "Risk assessment in a chemical plant with a CPN-HAZOP tool," *IFAC-Papers OnLine*, vol. 51, no. 11, pp. 939-944, 2018.

[3] I. V. Artamonov, A. P. Sukhodolov, "CPN tools-based software solution for reliability an analysis of processes in microservice environments," *International Journal of Simulation: Systems, Science and Technology*, vol. 19, no. 6, pp. 1-8, 2018.

[4] Y. L. Bai, X. M. Ye, "An improved CPN-based attacker model of cryptographic protocol," *Journal of Inner Mongolia Agricultural University*, (Natural Science Edition), vol. 35, no. 5, pp. 103-136, 2014.

[5] A. Baskar, R. Ramanujam, S. P. Suresh, "Dolev-Yao theory with associative blindpair operators," in *Proceedings of International Conference on Implementation and Application of Automata*, pp. 58-69, 2019.

[6] P. Birnstill, C. Haas, D. Hassler, J. Beyerer, "Introducing remote attestation and hardware-based cryptography to OPC UA," in *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'17)*, Limassol, Cyprus, pp. 1-8, 2017.

[7] C. Cremers, M. Dehel-Wild, K. Milner, "Secure authentication in the grid: A formal analysis of DNP3 SAv5," *Journal of Computer Security*, vol. 27, no. 2, pp. 203-232, 2019.

[8] D. Dolev, A. Yao, "On the security of public key protocols," *IEEE Transcations on Information Theory*, vol. 29, no. 2, pp. 198-208, 1983.

[9] Y. Gong, Z. Wang, D. Han, "OPC UA information modeling method and xml definition," in *IEEE Conference on Telecommunications, Optics and Computer Science (TOCS'20)*, Shenyang, China, pp. 328-331, 2020.

[10] R. Huang, F. Liu, D. Pan, "Research on OPC UA security," in *5th IEEE Conference on Industrial Electronics and Applications*, Taichung, Taiwan, pp. 1439-1444, 2010.

[11] V. M. Igure, S. A. Laughter, R. D. Williams, "Security issues in SCADA networks," *Computer Security*, vol. 25, no. 7, pp. 498–506, 2006.

[12] G. Karthikeyan, S. Heiss, "PKI and user access rights management for OPC UA based applications," in *IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA'18)*, Turin, Italy, pp. 251-257, 2018.

[13] F. Kohnhäuser, D. Meier, F. Patzer, S. Finster, "On the security of IIoT deployments: An investigation of secure provisioning solutions for OPC UA," *IEEE Access*, vol. 9, pp. 99299-99311, 2021.

[14] Z. R. Konigsberg, "Modeling and verification analysis of a flexible manufacturing system: A model logic approach," *Neural, Parallel & Scientific Computation*, vol. 26, no. 1, pp. 64-74, 2018.

[15] L. Li, F. Basile, Z. Li, "An approach to improve permissiveness of supervisors for GMECs in time petri net systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 237-251, 2019.

[16] Z. Luo, X. Zhang, "Research on OPC UA security encryption method," in *IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA'20)*, Chongqing, China, pp. 287-292, 2020.

[17] S. Marksteiner, "Reasoning on adopting OPC UA for an IoT-enhanced smart energy system from a security perspective," in *IEEE 20th Conference on Business Informatics (CBI'18)*, Vienna, Austria, pp. 140-143, 2018.

[18] S. G. Mathias, S. Schmied, D. Grossmann, R. K. Müller, B. Mroß, "A compliance testing structure

for implementation of industry standards through OPC UA," in *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'20)*, Vienna, Austria, pp. 1091-1094, 2020.

[19] N. Mühlbauer, E. Kirdan, M. O. Pahl, G. Carle, "Open-source OPC UA security and scalability," in *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'20)*, Vienna, Austria, pp. 262-269, 2020.

[20] S. C. Patel, G. D. Bhatt, J. H. Graham, "Improving the cyber security of SCADA communication networks," *Communication of ACM*, vol. 52, no. 7, pp. 139-142, 2009.

[21] M. Puys, M. L. Potet, P. Lafourcade, "Formal analysis of security properties on the OPC-UA SCADA protocol," in *International Conference on Computer Safety, Reliability, and Security*, pp. 67-75, 2016.

[22] A. Rashid, U. Siddique, S. Tahar, "Formal verification of cyber-physical systems using theorem proving," in *Proceedings of International Workshop on Formal Techniques for Safety-Critical Systems*, pp. 3-18, 2019.

[23] M. Rocchetto, N. O. Tippenhauer, "CPDY: Extending the Dolev-Yao attacker with physical-layer interactions," in *Proceedings of International Conference on Formal Engineering Methods*, pp. 175-192, 2016.

[24] M. Simon, D. Moldt, D. Schmitz, *et al.*, "Tools for curry-coloured petri nets," in *Proceedings of International Conference on Applications and Theory of Petri Nets and Concurrency*, pp. 101-110, 2019.

[25] A. Volkova, M. Niedermeier, R. Basmadjian, H. de Meer, "Security challenges in control network protocols: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 619-639, 2019.

Biography

Feng Tao, was born in 1970, researcher/PhD supervisor, CCF senior member, IEEE and ACM member.He graduated from Xidian University,and then worked at school of computer and communication in Lanzhou University of Technology. His research interests include Cryptography and information security.

Ma Zhuang-yu, was born in 1997,CCF member.He is a master's student at lanzhou university of technology.His research interests include technical information security and industrial control systems.

Fang Jun-li, was born in 1985,CCF member.She is a doctor's student at lanzhou university of technology.Her research interests include network and information security.