ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

DL/T698.45 Convergence Optimization Technology Design of Application Layer Service Extension Scheme and DLMS Application Layer Service Protocols

Zhongxing Wu[1,*], *Xuan Liu*[1], *Angang Zheng*[1], *Haotian Wang*[1], *Zhijun Xiang*[2] *and Zhongning Cui*[2]

[1] China Electric Power Research Institute Co., Ltd., Beijing, 100192, China
[2] China Electric Power Equipment and Technology Co., Ltd., Beijing, 100052, China

SUMMARY: *This paper first introduces the object-oriented DLMS/COSEM protocol and its characteristics, and then introduces the power information collection and management system-object-oriented interoperability data exchange protocol (DL/T698.45), which specifies the content of the master station, terminals and information transmission of the power information collection and management system. Aiming at the problem that the existing object-oriented protocol is time-consuming for terminal data collection, this paper designs a multi-source heterogeneous communication protocol that integrates the DLMS/COSEM protocol and the DL/T698.45 communication, and adds the corresponding attributes and methods in each functional module to realize the support for multi-source heterogeneous data communication in different functional modules. After that, the performance of the fused service protocols is tested using Peach fuzzy test method and adaptive genetic algorithm, combined with use case analysis. The results show that the Peach fuzzy tester proposed in this paper can shorten the test duration by 25.35% and reduce the number of test cases by 13.74%, which can effectively reduce the failure rate of test cases and make the model in this paper have a high test coverage and very good relevance. The average response time of the DLMS/COSEM-DL/T698.45 protocol is less than that of the DLMS protocol. The transmission success rate is significantly higher than that of the DLMS protocol, which shows that the design of the protocol that integrates the DLMS/COSEM protocol and the DL/T698.45 communication is more reasonable.*

KEYWORDS: *DLMS/COSEM protocol; DL/T698.45; Multi-source heterogeneous communication protocol; Peach fuzzy test tool; Adaptive genetic algorithm*

1 Introduction

In order to reduce human errors, lower operating costs, obtain optimal business benefits, and improve the comprehensive use of equipment and application systems, it is necessary to continuously extend and integrate existing smart grid functions to meet the needs. Currently, several more mainstream protocols are used in China, such as the DL/T645 standard protocol [1] and the IEC62056 international standard protocol [2], both of which have their advantages. DL/T645 standard protocol is easy to get started, simple and easy to understand, but for the gradual intelligent performance of the power meter great limitations [3]. IEC62056 international standard protocol system is huge, the statute is complex, has a strong interoperability, but for the Chinese power meter puts forward greater requirements and

*unceasingvigor@163.com
https://doi.org/10.65102/is2026281
Published: 30 April, 2026

challenges [4]. The Object-Oriented Data Exchange Protocol (DL/T698.45), is a set of interoperability protocols for acquisition systems based on an object-oriented modeling approach. Object-oriented technology has a wide range of application areas, such as product configuration [5], parking planning [6], and image analysis [7], etc. It is characterized by assemblability, extensibility, and reusability, and it solves the problems of data collection, data transmission, and system integration of billing devices [8]. DL/T698.45 adopts object-oriented technology, fully inherits the advantages of IEC62056 protocol framework, summarizes the traditional domestic communication statutes and application experience, and combines the new needs of the current smart meter. DL/T698.45 synthesizes the advantages and disadvantages of the existing protocols, and innovatively designs the protocols suitable for domestic AMI. The realization of DL/T698.45 not only slows down the pressure of on-site operation and maintenance of the power meter, but also promotes the standardization of the meter statute in a certain sense.

In recent years, many scholars have studied the security requirements of smart grid systems, such as the National Institute of Standards and Technology (NIST) and the International Electrotechnical Commission (IEC), in addition to the American National Standards Institute (ANSI) C12 series, which standardizes smart grid communications [9-11]. However, these two standards do not provide strong confidentiality as their application layer protocols allow AMI servers to access private meter data using AES decryption. According to a study, IEC61107 is a widely used communication protocol in the European Union, which has since been replaced by the IEC62056 standard, i.e., the DLMS/COSEM protocol [12]. The DLMS/COSEM protocol is an important communication protocol for electric power metering systems, which defines the format of the data exchange and the rules of communication between the meter and the concentrator and the master station, enabling the devices produced by different vendors to can communicate with each other to realize data acquisition, transmission and processing [13-15].

However, with the rapid development of information technology, network attacks are becoming more and more complex and diversified, the DLMS/COSEM protocol is facing more and more security threats, and malicious attackers may take advantage of the protocol loopholes to carry out behaviors such as illegal access, data theft or destruction, which will have a serious impact on the stable operation of smart grids and data security [16-19]. The DLMS defines a protocol for accessing (reading/writing) energy objects specified by the COSEM-specified commands for the application of energy objects and the function of communication using various communication media such as lines, which is similar to the language used for exchanging information in daily use, although various communication media such as the Internet, messenger and telephone are used in daily life [20-21]. When using DLMS to refer to data objects, the Object Identification System (OBIS) is used as an identifier to recognize these objects.COSEM is an energy object that contains metrological information (attributes) and functionality (methods) of energy, which can be represented as a separate object.COSEM as defined in the DLMS/COSEM standard has important functionalities that allow data storage of important information, responsible for access control and management, and time and event control as well as having payment functions [22-23].

The close integration of DLMS and COSEM makes the DLMS/COSEM standard protocol has a wide range of application prospects in the field of intelligent management on energy sources such as power and water resources. This protocol plays a vital role in the construction of smart grid, which guarantees the accuracy and real-time of power metering data, and provides an important basis for scheduling and decision-making of the power system [24]. Based on the above background, the convergence and optimization of DL/T698.45 application service extensions and DLMS/COSEM application layer service protocols are of great significance to

the study of smart grid security.

This paper first describes the concept and composition of the DLMS/COSEM protocol, after which it introduces the DL/T698.45 communication statute to optimize the model's access service and service model in combination with the IEC62056 application layer protocol and Q/GDW1376.1 link layer. Drawing on IEC62056's COSEM interface class concepts and OBIS models as well as requirements to innovate, highlighting the model's efficient and practical characteristics. Aiming at the problems of high redundancy in generating test cases and the inability to adjust the generation of fuzzy test cases according to the feedback of PLC under test in traditional protocol fuzzy testing, we construct the Peach fuzzy testing framework and establish a test case queue to save the fuzzy test cases that have been sent and the anomaly code corresponding to that test case, and calculate the individual fitness in genetic algorithms from the similarity of the individual to the seed in the seed queue and the anomaly code of that seed. The similarity between the individual and the seed in the seed queue and the anomaly code of the seed are calculated in the genetic algorithm, so as to realize the goal of adjusting the fuzzy test case generation according to the feedback of the PLC under test. Finally, the system is analyzed in terms of use case generation method, performance and application effect.

2 Design of protocol convergence optimization technique based on DL/T698.45 and DLMS

2.1 DLMS/COSEM protocols

2.1.1 Overview of DLMS/COSEM protocols

Traditional communication protocols for remote meter reading systems are generally linear in structure, with a single domain of use and serious shortcomings in terms of compatibility and interoperability.DLMS UA has developed a set of interoperable communication protocols: the Device Language Message Specification, DLMS, and the Companion Protocol for Energy Measurement, COSEM, which attempts to respond to the needs of the entire range of measurement meters and remote meter reading systems with a single specification that is compatible, independent, and scalable. Achieve protocol compatibility, independence, and extensibility. The protocol has additional features based on the characteristics of those past protocols, i.e., it enables the exchange of data for a wide range of energy types, such as electricity, heat, water, and gas. It adopts an object-oriented design concept to establish a unified logo/interface/service model from a communication perspective. Its applicability is strong, does not depend on the underlying communication medium, only based on the application layer protocol, easy to transplant and maintain, strong scalability, vendors in addition to the standard object interface can increase the data interface with self-describing characteristics, which can realize the interoperability of different vendors' equipment.

DLMS/COSEM communication protocol standard has been adopted by the IEC as an international standard, which has the following characteristics:

1) the use of object-oriented thinking, the definition of the information model of energy consumption instrumentation COSEM object model, with a standardized way of defining various types of data;

2) Its communication mechanism is based on the client/server approach, where the meter is the server and the meter reading host is the client;

3) The same object model is applicable to different meter types;

4) It has the characteristics of self-analyzing and interoperating;

5) The protocol layer and the physical layer are separated from each other;

6) Can be transmitted based on different media, such as RS485, Ethernet, GPRS, etc;
7) Applicable to a wide range of energy types, e.g. water, electricity, gas, heat, etc.

2.1.2 Components of the DLMS/COSEM protocol

The structure of DLMS/COSEM is shown in Fig.1. The IEC62056 standard system is divided into COSEM, which is a technical specification for power metering independent of communication protocols and media, and DLMS, which is a specification for equipment language messages based on the OSI reference model and IEC61334, and contains two parts, IEC62056-61 and IEC62056-62. DLMS has strong system interoperability and interoperability, not only integrates all the normative standards of electricity, water, gas and heat, but also supports a variety of communication media access methods.The DLMS/COSEM protocol adopts the Enhanced Performance Architecture EPA, which is a simplified version of the 7-layer OSI model, with only the Application Layer, the Data Link Layer (HDLC) and the Physical Layer.

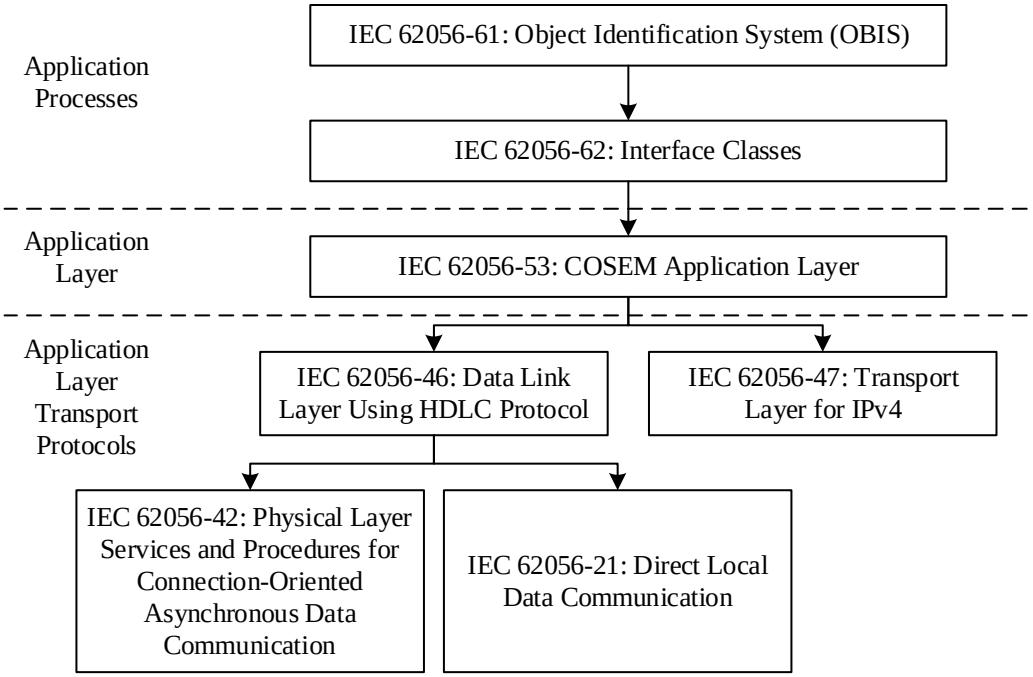


Figure 1: Structure chart of DLMS/COSEM

2.2 Object-oriented interoperable data exchange protocols

2.2.1 Scope of application of DL/T698.45 communication protocols

This part applies to the communication data exchange between the master station, collection terminal and power meter using point-to-point, multi-point common line and point-to-multi-point communication methods. Taking DL/T698.45 as an example, there is a lack of flexibility in specifying the operation of collection data items and collection methods. Therefore, this paper proposes a communication protocol design method based on object-oriented interoperability technology to optimize its data exchange protocol.

2.2.2 Communications architecture

Any communication network has four functions: information transmission, information processing, signaling mechanisms and network management. When planning, a unified communication protocol standard is followed in order to connect different operating systems and different hardware.

(1) Information exchange model

The information exchange model is divided into three kinds: direct exchange, request exchange and network platform exchange. When the master station accesses the collection terminal, the collection terminal is the server and the master station is the client; when the master station accesses the power meter, the power meter is the server and the master station is the client; when the collection terminal accesses the power meter, the power meter is the server and the collection terminal is the client. Data exchange between the client and the server, through the protocol can be real-time data collection, the physical channel is generally by the communication line, etc. for direct operation. This part of the information exchange model is shown in Figure 2.

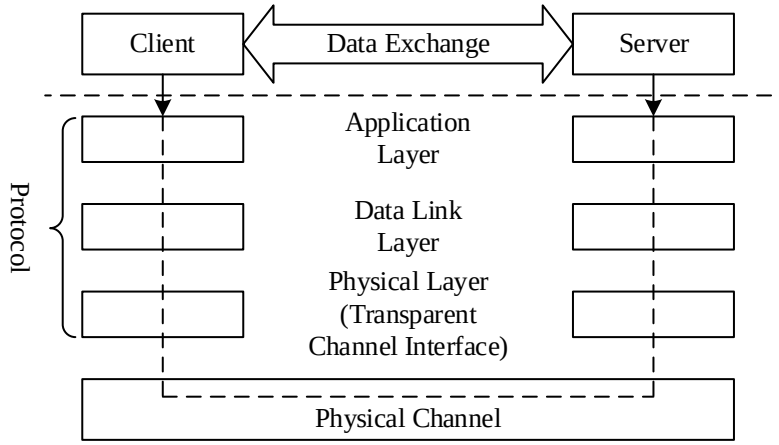


Figure 2: Information Exchange Model Diagram

(2) Application-oriented connection data exchange

Data exchange is done by establishing a dedicated communication route between two stations through nodes in the network. The process of establishing a connection between a client and a server is accomplished through an exchange line.

(3) Request/response type data exchange

The object-oriented protocol supports the request/response type of data exchange. When a client application process makes a service request to a server application process, the server application process provides a remote service response to the client application process.

(4) Notification/Acknowledgement Type Data Exchange

The object-oriented protocol supports the notification/confirmation type of data exchange, i.e., the server application process provides remote active reporting data service to the client application process according to the client's pre-customized active reporting content, and the client application process replies to the server application process with service confirmation.

(5) Client/Server Model

The client/server model is the basis of all network applications, i.e., the client actively initiates communication requests and the server passively waits for communication to be established. Physical devices are physically present, while logical devices are dependent on physical devices for their existence.

2.2.3 Data link layer

(1) Frame Structure and Frame Format

Frames are composed of multiple units that perform different functions to facilitate transmission. Object-oriented protocols use asynchronous transmission frame structure, the use of variable-length server address, used to achieve compatibility with a variety of meters. The frame header check HCS and frame check FCS use CRC-16 cyclic checksum and provide a unified and efficient algorithm.

(2) Length field L

Length domain L consists of 2 bytes, user data length: consists of bit0 to bit13, using bin coding, bit0 to bit13 is the length of user data, bit14 to bit15 is reserved. It is the number of frame bytes in the transmission frame except the start character and end character.

(3) Control field C

Control field C is 1 byte and is used by bit or combination of bits.

(4) Address field A

Address field A consists of the server address SA with a variable number of bytes and the client address CA with 1 byte. The server address consists of the address type, logical address, address length N, and its N byte address.

(5) Transmission rules

The transmission rules include:

a) When serial communication is used to realize local data transmission, 4 FEH are added as leading codes before the valid data frame when sending data.

b) The line idle state is binary.

c) There is no line space interval between the characters of the frame; the line idle interval between two frames requires a minimum of 33 bits.

d) If an error is detected according to e), the minimum line idle interval between two frames is 33 bits.

e) Frame header check HCS and frame check FCS.

f) Receiver checksum. If one of the checksums fails, the frame is discarded; if there is no error, the frame data is valid.

(6) Framing rules

When the length of a complete application layer protocol data unit exceeds the maximum size of the sending frame, it can be transmitted in frames. Split-frame data receiver should confirm the split-frame transmission line by line, using split-frame transmission, the control domain in the split-frame flag position 1. The client's server receives the process, it will be split-frame operation, according to the length of the frame to determine the split-frame flag.

2.2.4 Data application layer

(1) Data application layer services

The data application layer service object includes three necessary components: pre-connection, application connection and data exchange. Pre-connection service is applicable to exchange network transmission channel, such as Ethernet, GPRS, etc. When it completes the physical connection and establishes a transparent channel, it is necessary to establish pre-connection on this channel and manage it.

(2) Application layer data unit specification

The basic rules of Application Protocol Data Unit (APDU) follow the abstract syntax of ASN.1, and the details of this syntax can be referred to GB/T16262.1-2006.The following is a brief description of the important data identification.

1) PIID. The APDU serial number with ACD flag bit and the priority flag PIID-ACD are used in each service data type of the client APDU for each service data type of server APDU.

The server is sorted by priority into general and advanced.

2) Object Attribute Descriptor OAD. bit0...bit7 is used to represent the lowest to the highest bit of the octet, where: bit0...bit4 encodes the object attribute number; bit5...bit7 encodes the attribute characteristics.

3) Object Method Descriptor OMD. Is used to describe the method of the object. Generally there are many methods, in the program we can choose a certain method to describe the object. This purpose can be achieved by setting OMD.

4) Data security MAC is particularly important in energy meters, where data security plays a vital role for the customer as well as subsequent operational functions.

2.2.5 Objects and object identification

The DL/T698.45 protocol utilizes object-oriented thinking and treats physical quantities, event records, freeze records, etc. as objects. Event record, freeze module has similar functions, all belong to the event, freeze type data, and large amount of data, but also the most complex, the most attributes, the largest amount of data, the most need to realize the method of the DL/T698.45 protocol module.

2.3 Multi-source heterogeneous communication protocol design based on the convergence of DL/T698.45 and DLMS/COSEM

2.3.1 Overall design of multi-source heterogeneous communication protocols

The multi-source heterogeneous communication protocol that integrates DLMS/COSEM protocol and DL/T698.45 communication belongs to the application layer protocol and is designed to improve and optimize the business functions. The development of Multi-source Heterogeneous Communication Protocol adopts object-oriented method, follows the specification of object-oriented interoperability data exchange protocol, extends the existing object-oriented protocol, improves the efficiency of business execution on the basis of meeting the business requirements, and at the same time, it can be convenient for the reuse, modification and extension of each functional module.

Multi-source heterogeneous communication protocol is applicable to the communication of each functional module of modularized terminal, which usually includes delivery module, carrier module, RS 485 module, communication module and remote pulse module. The communication module meets the communication between the master station and the concentrator, and the existing communication protocol mainly promotes the DL/T698.45 object-oriented protocol, which contains the data and events of the concentrator body, the data and events of the measurement points collected by the concentrator, and the format of the data and events of the measurement points are propagated according to the DL/T698.45 object-oriented protocol. The implementation of the multi-source heterogeneous protocol that integrates the DLMS/COSEM protocol and the DL/T698.45 communication is achieved by expanding on the existing object-oriented protocols, which effectively avoids the growth of the code processing workload.

2.3.2 Multi-source heterogeneous communication protocol module implementation

The main extensions of the multi-source heterogeneous communication protocol to the object-oriented protocols are shown in Table 1. the RS 485 module includes two interfaces, the gate node input interface and the alarm output interface. the RS 485 interface is used to collect the energy meter data, the gate node input is used to collect the gate node status information, and the alarm output is used to send an alarm message before the terminal issues a trip-closing

control command to the user. the RS 485 module needs to be extended to the The RS 485 module needs to be expanded for F224 and F223.

The telecom pulse module includes two telecom input interfaces and two power meter pulse input interfaces, and the telecom inputs and pulse inputs are both in 0 - 1 state. This module needs to expand F222 and F20A, and attribute 3 of F20A provides the number of pulses and related time scale base data, which provides data support for electric energy calculation. The communication module includes remote communication and local communication. The remote communication module is used for the communication between the master station and the terminal, which adopts GSM, GPRS, CDMA, etc., and needs to expand its F221 object. The local communication module adopts power line carrier communication or civil radio special frequency band communication, which is used to communicate with the collector and the meter, and needs to expand its F209.

The control module mainly forwards and processes the control commands of the display module to the user, such as trip and close, so as to cooperate with the completion of the terminal's control function, which requires the expansion of F220, F205 and F203.

Table 1: Extension of objects

Target	Extended properties and methods
F224 Door Node Status	Property 2 Switching Unit: Status ST, Change CD
	Property 4: Switching signal access flag and switching signal property flag
F223 485 Function Module	Property 2: Device Type and Function: Device Type, 485-1 Port, 485-2 Port, Alarm Relay, Pre-Action, Cycle Relay, Door Node
F222 Remote pulse collection module	Property 2 (same as F223 Property 2)
F20A Pulse Input Device	Attribute 3 Pulse Unit: Number of pulses, start 20 ms time scale, start minute time scale, end 20 ms time scale, end minute time scale
F221 Communication module	Property 2 (same as F223 Property 2)
F209 Carrier/micro-power wireless interface	Property 2 Local Communication Module Unit: port descriptor, communication parameters, version information
	Method 127 parameters: communication address, message timeout time for receiver, transparent forwarding command
F220 Control module	Property 2 (same as F223 Property 2)
	Attribute 3-round unit: gate status, switch attribute, wiring status, pre-start status
	Method 127 Modify switch properties (round number, gate status, pre-motion status)
F205 Relay output	Property 2 Relay Unit: Descriptor, Current State, Switch Property, Wiring State
	Method 127 Modify switch properties (relay number, switch properties)
	Method 128 Modify Status (Relay Number, Status)
F203: Digital input switch	Property 2 (same as F224 Property 2)
	Property 4 (same as F224 Property 4)

3 Protocol fuzzy test case generation methodology and performance analysis

3.1 Introduction to fuzzy testing and the Peach framework

As a test method used to explore the potential risk of the object, fuzzy test with its simple operation method, simple requirements, higher level of automation and test results are more reliable and other characteristics, and by many researchers in the field of industrial control and safety concerns, and Peach fuzzy test tool as a more representative and the use of the feedback is more desirable fuzzy test one of the examples of the application of the following fuzzy test and Peach fuzzy test tool to develop a lengthy overview of the following, respectively. The following is an overview of the fuzzy test and the Peach fuzzy test tool.

3.1.1 Overview of fuzzy testing

As an automated or semi-automated work of security risk detection techniques, fuzzy testing is an application security testing method based on defect injection, which sends a large amount of randomly generated data to the test object in an attempt to interfere with the test target, or even hope to cause the test target to crash, so as to successfully excavate the existence of the test target's security vulnerabilities, which usually include data coding errors, software code errors or hardware errors, or design flaws in operating systems and network protocols. Such security vulnerabilities usually include data coding errors, software code errors or hardware errors, or design flaws in operating systems and network protocols. The workflow of fuzzy testing can be summarized as follows: confirming the test target, constructing test cases, generating test data, executing fuzzy tests, monitoring the test target, and analyzing and confirming test results.

3.1.2 Overview of the Peach Fuzzy Testing Framework

According to the design idea of fuzzy test, Peach fuzzy test framework is formed by combining several parts, and different components are responsible for realizing different functional modules, which include format parser, fuzzy test engine, data mutator, data generator, test state changer and state monitor. These components are introduced in the following order.

(1) Format Parser. The format parser is responsible for verifying whether the format of the Pit file, which is used as a template for generating test data, conforms to the definition of the Peach fuzzy test framework, and if the verification is correct, the Pit file will be converted to the format of the DOM tree through parsing and compilation.

(2) Fuzzy test engine. According to the definition in the Pit file, the fuzzy test engine is used to wake up and start the Peach fuzzy test framework components specified in the Pit file in turn, and after the engine completes the initialization of the components, these components will be involved in the fuzzy test testing process.

(3) Data Mutator. According to the data format and mutation strategy specified in the Pit file, the fuzzy test engine will call different types of data mutators for fuzzy test data generation.

(4) Data Generator. The data generator will call the data mutator to generate fuzzy test data and selectively send different fuzzy test data according to the state change of the test target.

(5) Test state changer: Several states and operations corresponding to different states are predefined in the Pit file, and the test state changer is responsible for switching between different states according to the Pit file, and executing the corresponding operations after the state switching is completed.

(6) State Monitor. The test object may have different state changes during the fuzzy testing process, which requires the state monitor to keep monitoring the test object all the time during the testing process, accurately record the state changes of the test object in real time, and

feedback this change to the testers, so as to facilitate the testers to carry out the subsequent work related to the investigation of potential safety hazards.

The structure of Peach fuzzy test framework is shown in Figure 3, including fuzzy test dependency files, fuzzy test engine and monitoring test objects.

The test dependency files are mainly Pit files and Python library files that may need to be invoked during the test process, of which the Pit file is the key configuration file of the Peach fuzzy test framework, which is written in XML language, and the testers decide what way to execute fuzzy tests by writing the Pit file.

Peach fuzzy test engine mainly includes the following components, including Peach Engine, Agent, Strategy, Mutator, Publisher and Logger, etc. Peach fuzzy test framework has its own set of logging system, and supports the user to customize the extension according to the actual needs of the log is used for recording The log is used to record the data sent during the fuzzy testing process and information about the state changes of the test object.

In order to realize the needs of fuzzy testing on different operating systems and better monitor the state changes of test objects on different platforms, Peach defines Windows monitor, Linux monitor, OSX monitor, and cross-platform monitor, etc. to facilitate users to conduct fuzzy testing on different platforms.

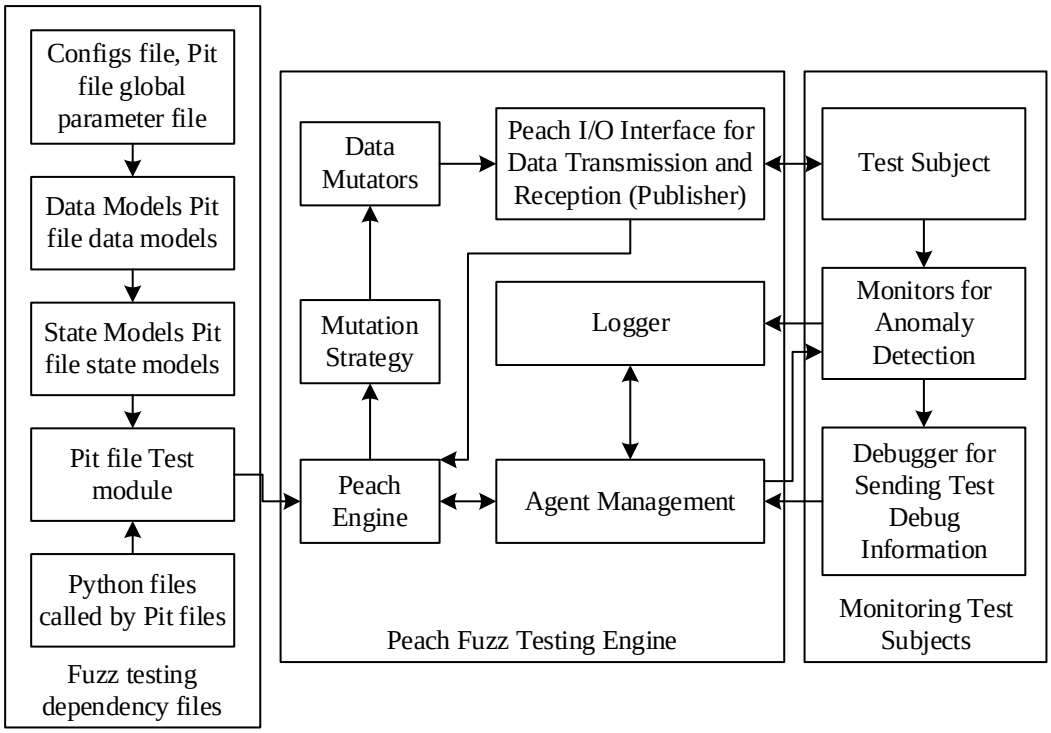


Figure 3: The Basic Structure of Peach Fuzzy Testing

3.2 Genetic Algorithm Based Protocol Fuzzy Test Case Generation Approach

3.2.1 Coding

The purpose of coding is to convert the parameters of the problem space into a form that can be handled by the genetic algorithm, and the method of coding will have a direct impact on the next few steps such as crossover. Therefore, choosing the appropriate coding method will effectively improve the computational efficiency. Common coding methods include: binary

coding and Gray coding and so on. Binary coding is a frequently used coding method in genetic algorithms. Its disadvantage is that it is not applicable to multi-dimensional, high-precision continuous function optimization problems, and there may also exist a large Hamming distance, also known as the Hamming cliff, so this paper chooses Gray coding method.

3.2.2 Adaptation function

In this paper, for the traditional Modbus TCP protocol fuzzy test method, the evolutionary direction of the genetic algorithm is adjusted in time, the returned anomaly code is used to indicate the code coverage of the test case, and different weights are assigned to different anomaly codes, the maximum weight is w_4 when the anomaly code is 4, followed by anomaly code 2, anomaly code 3 and anomaly code 1, which are assigned the weights of w_2, w_3 and w_1, if the generated test case is responded normally by the industrial control device, i.e., the constructed test case belongs to the normal packet, the test case is assigned the minimum score of w_0. i.e:

$$w_0 < w_1 < w_3 < w_2 < w_4 \tag{1}$$

First, a test case queue is created to store the best test case calculated by the genetic algorithm each time and the exception code in the response message of the PLC under test after receiving this test case. The initial queue consists of Modbus TCP protocol packets captured by the host computer during normal communication with the PLC.

Then, after generating the individual p_x, all the individuals in the test case queue with the same function code as the individual p_x are formed into a set P_y. Calculate the distance between p_x and each byte of the individual p_j in the set P_y, and then add up the distances of each byte to calculate the distance between p_x and the individual p_j, and after that, add 1 to the calculated distance, where 1 is added in order to avoid the distance to be 0. Take the inverse of the number of times and multiply it by the weight w_j which is calculated based on the anomaly code of p_j. i.e:

$$\begin{aligned} &D\left(p_x, p_j\right) = \frac{w_j}{\left(\sum_{i=0}^{l_j-1}\left(b_i^x - b_i^y\right)^2\right)+1} \\ &w_j \in \left\{w_0, w_1, w_2, w_3, w_4\right\} \\ &B\left(p_x\right) = \left(b_0^x, b_1^x, \cdots, b_{l_j-1}^x\right) \\ &B\left(p_j\right) = \left(b_0^y, b_1^y, \cdots, b_{l_j-1}^y\right) \end{aligned} \tag{2}$$

After that, the distance between p_x and each individual in the set P_y is added and divided by the number n of the set P_y to give the average distance between an individual p_x and the initial population:

$$\overline{D\left(p_x, P_y\right)} = \frac{\sum_{j=0}^{n-1} D\left(p_x, p_j\right)}{n}, P_y = \left(p_0, p_1, \cdots, p_{n-1}\right) \tag{3}$$

Finally, the final calculated mean distance is used as the fitness value for that individual:

$$fitness\left(p_x\right) = \overline{D\left(p_x, P_y\right)} \tag{4}$$

3.2.3 Selection

After completing the calculation of individual fitness value in the previous step, in order to select individuals with good genes to continue to survive and reproduce, this paper uses the "roulette wheel selection operator" to select two parents from the parent population.

The calculation steps of the roulette operator are as follows:

Step 1: Calculate the fitness value of an individual according to the formula, and calculate the probability of all individuals in the population being inherited $\Pr o\left(msg_i\right)$:

$$\Pr o\left(msg_i\right) = \frac{fitness\left(msg_i\right)}{\sum_{j=1}^{N} fitness\left(msg_j\right)} \tag{5}$$

Step 2: Calculate the cumulative probability Q_i for all individuals in the population:

$$Q_i = \sum_{j=1}^{i} \Pr o\left(msg_j\right) \tag{6}$$

Step 3: After that, random values $rand$ are generated in the interval $[0,1]$. If $r < Q_1$, then select individual 1, otherwise select individual k that satisfies Eq. (7) and enter the selected individuals into the offspring population for the next step:

$$Q_{k-1} \le rand < Q_k \tag{7}$$

Step 4: Repeat step 3 over and over again to get individuals that form a new population.

3.2.4 Crossing and variation

The crossover operators applicable to binary coded individuals include single-point crossover and two-point crossover. In this paper, we choose to adopt single-point crossover, i.e., we randomly select a site on the binary gene sequences encoded in the two parent individuals, and exchange the gene sequences of the two parents at the back of this crossover site in order to generate a new next-generation individual.

Mutation This paper takes a simpler implementation. Since the encoded gene sequences are all binary data strings, the basic bitwise variation operator is used in this paper.

In order to retain individuals with high fitness values and eliminate those with low fitness values, this paper uses an adaptive genetic algorithm to compute the crossover probability P_c and the variance probability P_m. That is:

$$P_c=\begin{cases}\dfrac{k_1(f_{\max}-f)}{f_{\max}-f_{avg}}, & f\geq f_{avg}\\ k_2, & f<f_{avg}\end{cases}, P_m=\begin{cases}\dfrac{k_3(f_{\max}-f')}{f_{\max}-f_{avg}}, & f'\geq f_{avg}\\ k_4, & f'<f_{avg}\end{cases}$$
$$k_1,k_2,k_3,k_4\in[0,1] \tag{8}$$

where $f_{\max}$ denotes the maximum of the fitness values of all individuals in the population; f_{avg} denotes the average of the fitness values of all individuals in the population; f denotes the larger of the fitness values of the two individuals that will crossover; and f' denotes the fitness value of the individual that is going to be mutated.

3.3 Performance Analysis of Use Case Generation Methods

In order to validate the effectiveness of the proposed method and test its performance, the open source fuzzy test method Peach and the method of this paper were used to test the target respectively. The main identical parameters in the two sets of experiments are set as follows: population size $size=100$; crossover standard probability $p_c=0.75$; and variation standard probability $p_m=0.05$.

The termination condition of the experiment is to dig out the vulnerability. Firstly, in order to test the performance of the use case optimization method proposed in the previous paper, the use cases generated by the two methods are counted in terms of population as the average similarity of use cases in each generation, where the average similarity of use cases is calculated as:

$$s=1-\frac{1}{n\times d_{\max}}\times\sum_{i=1}^{n}d_{i-case} \tag{9}$$

3.3.1 Trends in average similarity of use cases

The trend of the average similarity of the use cases is shown in Figure 4. After analyzing the results, it can be seen that in GA-Fuzzer test, the variation probability is not adjusted by judging the state of the use case generation after it is given by the configuration file before the test, and although the average similarity of the use cases is lower, the targeting is weaker, and the generation of the use cases tends to be random. While the state of Peach's use case population is dynamically adjusted, when the number of danger points increases, due to the characteristics of Peach's fuzzy test, it will lead to the gradual convergence of the generated use cases to the danger points to which they belong, and at this time, the appropriate adjustment of the variance probability can improve the vulnerability hit rate near the danger points. In addition, in the results there are some populations whose use case similarity exceeds 0.7, for example, the population around 754 generations, after analyzing the original data, the case is the existence of danger points within the population, and tends to converge. According to the dynamic fitness function, the use case generation at this time tends to converge to the danger point, which improves the vulnerability hit probability of the use cases, and therefore the similarity between the use cases is high.

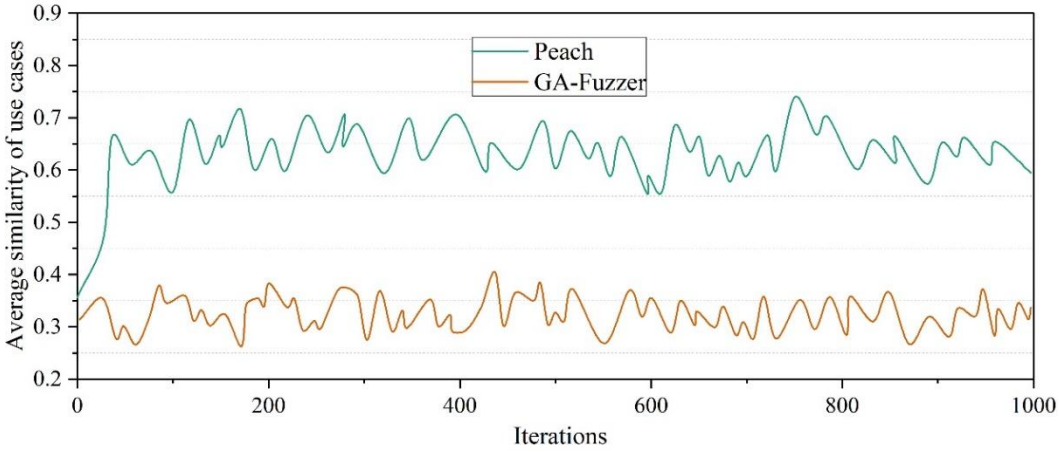


Figure 4: Trend of average similarity change in use cases

3.3.2 Distribution of use cases generated by each method in the use case space

After the two groups of experiments were tested several times in the same environment respectively, the generated use cases were counted and the average of the test data of the use cases generated by each method is shown in Table 2. The results show that the Peach fuzzy tester proposed in this paper has an advantage in time efficiency and can shorten the test duration by 25.35% compared to GA**-**Fuzzer test tool in the same experimental environment with successful completion of the test.

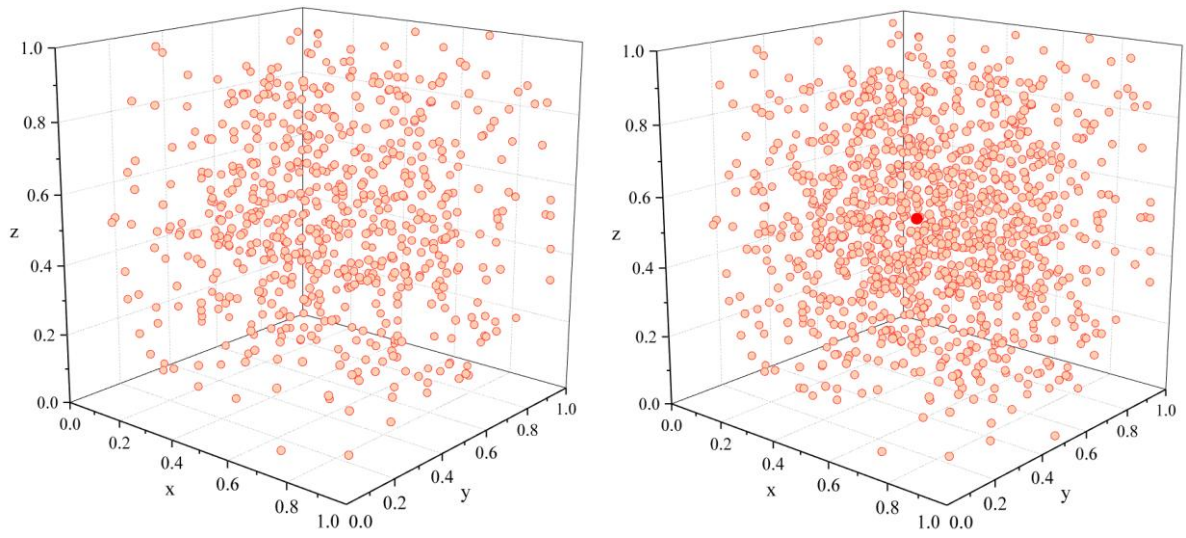
Table 2: The average of test case data generated by each method

Generative approach	Number of use cases	Similarity	Duration /h	Number of vulnerabilities
GA**-**Fuzzer	653	0.4159	0.6647	0.9
Peach	1138	0.5778	0.4962	1.00

Meanwhile, representative consecutive 100-generation populations are selected as example data in the three groups of tests, which are normalized and abstracted into the three-dimensional use-case space, and the distribution of use cases generated by each test method in the use-case space can be obtained, and the distribution of the use cases generated by each method in the space is shown in Fig. 5, in which (a) and (b) represent the GA **-** Fuzzer and Peach algorithms, respectively, and red points represent the number of vulnerabilities. From the distribution state of the test cases of each testing method within the use case space, the use cases generated by the GA**-**Fuzzer fuzzy testing framework are more evenly distributed within the use case space. This is due to the fact that its use case generation does not have a better control strategy and tends more towards higher coverage. The method is able to maintain some variability between individuals, but is not well targeted to locate vulnerabilities quickly, which leads to a longer time required for testing and a large number of test cases.

There is one danger point in the example data, according to the design of the proposed method, when there is a danger point in the space of use cases, the individuals within the population will first cluster according to the danger points closer to them, and then each of them will genetically converge within their class, so that multiple suspected vulnerabilities can be targeted at the same time to improve the efficiency of the test; in the absence of a danger point in the space or at the end of the survival cycle of all the danger points, the The generation of use cases tends to have a high coverage and low similarity trend, ensuring that the test has a high coverage rate. By dynamically updating the fitness function of this paper's algorithm with

respect to the presence or absence of dangerous points in the space, it can simultaneously have high test coverage and good targeting, effectively reducing the failure rate of test cases.



(a)GA-Fuzzer (b)Peach

Figure 5: Distribution of use cases generated by each method in space

4 Analysis of the application effect of fuzzy test cases for protocols based on genetic algorithms

4.1 Experimental environment

The smart meter simulator in TCPWRAP-PER communication mode runs in a physical machine with an Intel Core i9-10900K processor, Windows Professional operating system, and 128GB of RAM; DSFUZZ is deployed as a whole in a computer-side virtual machine, Vmware, on an Ubuntu 18.04.6 LTS system. The virtual memory is 32GB.

4.2 Fuzzy test experiments in multiple modes

During the fuzzy testing of this experiment, the main statistics are the total number of test sequences, the total number of newly discovered states, and the total number of crashes. In this section, fuzzy tests with and without update mechanism are performed using Peach for HDLC-based and TCPWRAPPER-based communication models, respectively. The update mechanism refers in using the request message tree based DLMS/COSEM protocol proposed in this paper. Each fuzzy test was run for a total of 12 hours, and the data at runs up to 5, 8 and 12 hours are tabulated in the table.

(1) With update mechanism vs. without update mechanism

The results of fuzzy test in HDLC based communication mode are shown in Table 3. It can be found that the number of sequences generated and tested by the fuzzy test with the update mechanism is consistently higher than without the update mechanism, the number of states found and the number of crashes are equal to or higher than the fuzzy test without the update mechanism for the same period of time in the HDLC communication mode. The number of states and crashes found by both mechanisms is the same when running up to 5 and 8 hours, but during the period from 8 to 12 hours, the number of states and crashes found by the

mechanism with updating continues to increase, while the number of states and crashes in the mechanism without updating no longer increases. This means that for the fuzzy test without updating mechanism, the total number of states and crashes found at this time is close to the maximum number of states that can be found by this mechanism, while for the fuzzy test with updating mechanism, there is still a greater possibility that new states and crashes can continue to be found.

Table 3: Fuzzy Testing Results Based on HDLC Communication Mode

Test items	Running time/hour	Total number of test sequences	Total number of detected states	Total crashes
Update mechanism	5	798	1	0
	8	8636	7	1
	12	17005	11	2
No update mechanism	5	637	1	0
	8	7515	7	1
	12	14637	7	1

The fuzzy test results in TCPWRAPPER mode are shown in Table 4. The test results have a similar trend with comparing the fuzzy test results in HDLC communication mode, and the enhancement is more obvious. This indicates that the update mechanism can improve the fuzzy test in both HDLC communication mode and TCPWRAPPER communication mode.

A comprehensive analysis reveals that the speed of fuzz testing is higher in the TCPWRAPPER communication mode, and therefore more states and crashes can be found in the same amount of time. In the next experiments of this paper, we will use TCPWRAPPER communication mode for fuzzy testing in order to achieve better testing results in the same time.

Table 4: The Results of Fuzzy Testing in TCP Wrapper Mode

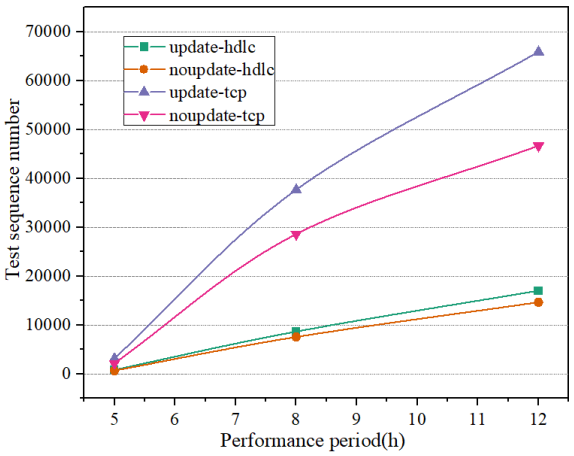
Test items	Running time/hour	Total number of test sequences	Total number of detected states	Total crashes
Update mechanism	5	3113	1	0
	8	37652	10	3
	12	65878	14	10
No update mechanism	5	2139	2	0
	8	28565	10	3
	12	46657	10	3

(2) HDLC communication mode vs. TCPWRAPPER communication mode

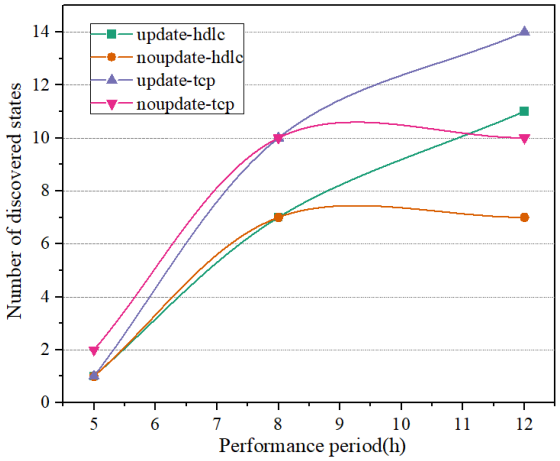
The visual data comparison results of the four modes are shown in Figure 6. Where the legends update-hdlc, noupdate-hdlc, update-tcp, and noupdate-tcp represent the HDLC mode with update mechanism, HDLC mode without update mechanism, TCPWRAPPER mode with update mechanism, and TCPWRAPPER mode without update mechanism, respectively; (a) ~ (c) represent the comparison results of the number of test sequences in different modes, the number of states found in different modes and the number of crashes triggered in different modes, respectively.

The results show that the speed of fuzzy testing in the test sequences generated in different modes with the update mechanism is higher than that without the update mechanism, and it is more obvious in the TCPWRAPPER mode. The trend of the number of discovered states over time in different modes shows that more states are discovered when the update mechanism is

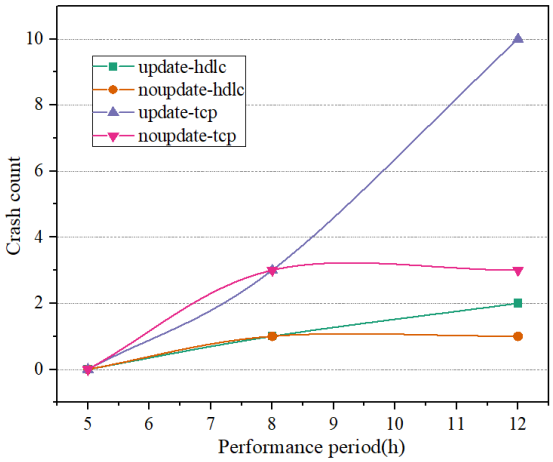
used, both in HDLC and TCPWRAPPER modes; and the results of the number of triggered crashes are similar to the trend of the number of discovered states, which is more when the update mechanism is used. This strongly proves the effectiveness of the update mechanism in the APDU variant algorithm for the DLMS/COSEM protocol based on the request message tree, and it shows good fuzzy test results both in the HDLC-based communication mode and in the TCPWRAPPER-based communication mode. The comprehensive analysis shows that the fuzzy test is most effective when there is an update mechanism in the variant algorithm and based on the TCPWRAPPER communication mode.



(a)Number of test sequences for different modes



(b)Number of states found in different modes



(c)Number of crashes triggered by different modes

Figure 6: Intuitive comparison of data for four modes

4.3 Comparative experiments with different fuzzy testing tools

This section compares Peach with existing fuzz testing tools Boofuzz, DSFUZZ and eFuzz. In the same experimental environment, the fuzzy test results of different tools are shown in Table 5. It can be found that DSFUZZ and eFuzz have the slowest testing speed, with 1854 and 1814 tests per hour, respectively, and the least number of states found and crashes triggered, with 5 states found and 2 and 3 crashes triggered, respectively. Boofuzz has the fastest test speed of

4034 tests per hour, and finds slightly more states and triggers slightly more crashes than DSFUZZ and eFuzz, at 7 and 5, respectively. Peach, the tool proposed in this paper, has a lower test speed compared to Boofuzz and a higher test speed compared to DSFUZZ and eFuzz, at 3,439 times per hour; however, Peach finds the most states and triggers the most crashes, with 16 states found and 9 crashes triggered.

During the 12 hours of operation, Boofuzz, DSFUZZ, and eFuzz found fewer states and triggered fewer crashes, and the test cases they generated were not only blind and did not conform to the coding rules used by DLMS/COSEM, but most of them were rejected by the smart meters, could not be responded to efficiently, and would take some time due to the waiting process, so the test speed is faster compared to DSFUZZ and eFuzz, even though Peach variants are more complex. The Boofuzz test is faster compared to the DSFUZZ and eFuzz tests because it does not include the Release connection release request in each test sequence, which is one-third less request message packets on average. The above experiments demonstrate that Peach has better fuzzing results than existing test pattern tools, has higher test speed, and can trigger more crashes in the same amount of time.

Table 5: Results of Fuzzy Testing with Different Tools

Tool	Test speed (times per hour)	Found status (count)	Triggered crash (times)
Boofuzz	4034	7	5
DSFUZZ	1854	5	2
eFuzz	1814	5	3
Peach	3439	16	9

4.4 Analysis of Vulnerability Detection Results

Comparison of DLMS/COSEM-DL/T698.45 and DLMS and DLMS/COSEM fuzzy test results are shown in Table 6. In this paper, the DLMS/COSEM-DL/T698.45 fusion protocol found the problem of multiple memory leaks in the code. Analyzing the error alerts given in the test results, it can be found that they are all caused by dynamically opening memory in the code and not releasing it correctly. In addition to the above memory leaks, after using Peach to perform fuzzy testing and analyzing the triggered crashes, this paper found a Use After Free vulnerability in the DLMS/COSEM-DL/T698.45 fusion protocol client. Protocol client using Peach fuzzy test can find multiple vulnerabilities in the code, which proves the effectiveness of the DLMS/COSEM-DL/T698.45 Fusion Protocol by comparing it with existing fuzzy testing tools.

Table 6: Comparison of the effects of fuzzy testing

Tool	Execution speed (times/s)	Code coverage (%)	Trigger crash (thousands)	The only crash
DLMS	137.0527	3.27	1.257	1
DLMS/COSEM	713.6418	9.65	12.709	18
DLMS/COSEM-DL/T698.45	668.9534	38.94	45.138	33

4.5 Performance comparison of DLMS/COSEM-DL/T698.45 convergence protocols

During the protocol testing, this paper sets the same network parameters respectively. Whenever 100 polling times are added, the average response time and transmission success rate of the two protocols are recorded. In the case of 0 to 5000 polling times, the average response time and

transmission success rate of the two protocols “electrical energy data” statistics.

(1) Average response time of protocols

The average response time of the two protocols is shown in Figure 7. The results show that when the number of polls is less than 5000, the average response time of both DLMS/COSEM-DL/T698.45 convergence protocol and DLMS protocol is less than 35 ms, which meets the protocol requirement for response time (<50ms). In addition, the average response time of the DLMS protocol is significantly higher than that of the DLMS/COSEM-DL/T698.45 fusion protocol. The main reason for this is that the checksum process does not occur in the DLMS/COSEM-DL/T698.45 fusion protocol, and compared with the DLMS protocol, the execution steps of the DLMS/COSEM-DL/T698.45 fusion protocol are fewer, which directly leads to the fact that the average response time of the DLMS/COSEM-DL/T698.45 fusion protocol must be smaller than that of the DLMS protocol.

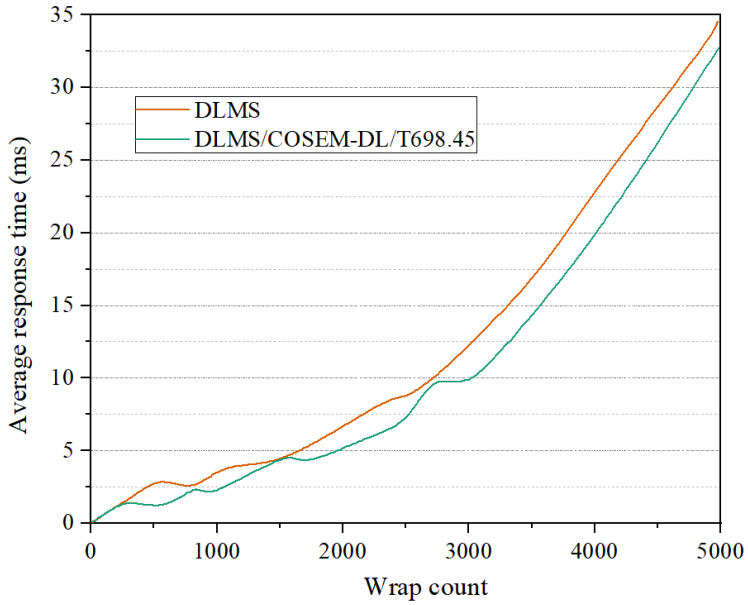


Figure 7: Average response time of two protocols

(2) Transmission success rate of protocols

The transmission success rate of different protocols is shown in Fig. 8. When the number of polling is greater than 2810, the transmission success rate of DLMS/COSEM-DL/T698.45 fusion protocol and DLMS protocol reaches 99.95% and 91.73% respectively, i.e., it meets the protocol requirement for transmission success rate (>90%). In addition, the transmission success rate of the DLMS/COSEM-DL/T698.45 convergence protocol is significantly higher than that of the DLMS protocol, and there is a large gap between the two, which is mainly due to the fact that the design of the DLMS/COSEM-DL/T698.45 convergence protocol is more reasonable, and the implementation of its protocols differs slightly, so that there is a large difference in its transmission success rate.

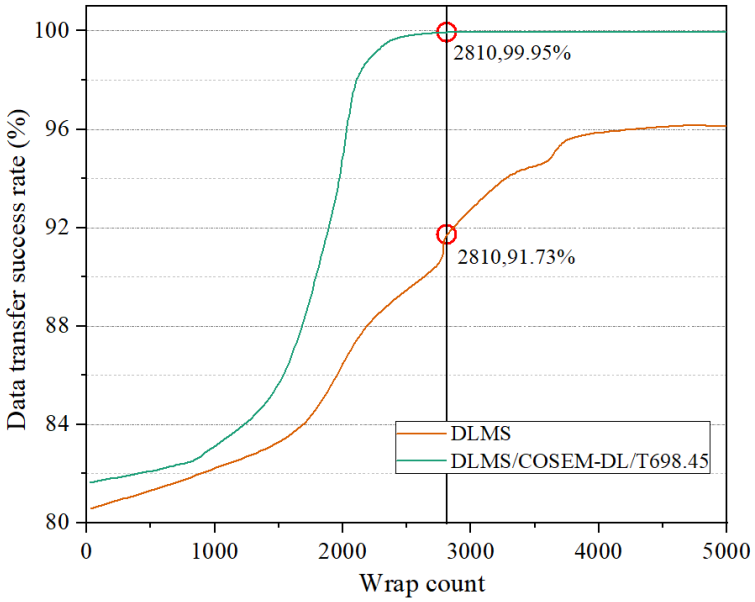


Figure 8: Success rate of different protocols

5 Conclusion

In this paper, the DLMS/COSEM protocol and DL/T698.45 communication are converged, and the performance of the converged service protocol is tested using an improved fuzzy testing method with Peach.

The improved Peach fuzzy testing tool can adjust the variation probability by judging the state of the use case generation, the average similarity of the use cases is lower, the targeting is stronger, and the generation of the use cases gradually converges to the danger point to which they belong, which improves the hit rate of the vulnerabilities near the danger point. The method can target test multiple suspected vulnerabilities at the same time to improve the testing efficiency; its generation of use cases tends to have a high coverage and low similarity trend, which effectively reduces the failure rate of test cases. Compared with existing test pattern tools, Peach has better fuzzy testing effect, has higher testing speed, and can trigger more crashes in the same time. The design of the converged service protocol is more rational, and the transmission success rate and average response time of this protocol are significantly better than that of the DLMS protocol, and there is a large gap between the two, so that the difference in their transmission success rates is larger.

Acknowledgements

The study was supported by “Science and technology project of the headquarters of SGCC ‘Key Technology Research on Electricity Information Collection for Overseas BOOT Mode’ (Project Number: 5700-202455266A-1-1-ZN)”.

References

[1] Guo, J., & Liu, D. (2012). Design of a Smart Meter Recorder with Mass Storage Based on DL/T645-2007 Protocol. In Internet of Things: International Workshop, IOT 2012, Changsha, China, August 17-19, 2012. Proceedings (pp. 660-666). Berlin, Heidelberg:

Springer Berlin Heidelberg.

[2] Drăgan, F., Holonec, R., & Copîndean, R. (2019, May). Local Monitoring/Recording and Display Device for Power Electricity Meter, using IEC 62056–21 Local AMR application device, hardware solution, for DLMS-COSEM based Power Meters. In 2019 8th International Conference on Modern Power Systems (MPS) (pp. 1-6). IEEE.

[3] Wang, J., Dai, Y., Sun, W., & Xu, X. (2011, September). Wireless sensor networks communication QoS-MAC model based on DL/T645 protocol. In 2011 International Conference on Electrical and Control Engineering (pp. 1407-1410). IEEE.

[4] Bo, Z., Baiyuan, Q., Gang, L., & Shuai, H. (2013, December). Communication architecture of interactive energy measurement based on IEC62056. In Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC) (pp. 444-447). IEEE.

[5] Wang, Y., & Ai, Q. S. (2011). Products information inheritance based on Object-Oriented technology. Advanced Materials Research, 328, 279-282.

[6] Ni, M., Sun, Z., Luo, Y., Yi, Q., Zhang, Y., & Wang, Z. (2021). Evaluation model of parking equipment planning and design based on object-oriented technology. Applied Sciences, 11(9), 4263.

[7] Mahmoudi, F. T., Samadzadegan, F., & Reinartz, P. (2013). Object oriented image analysis based on multi-agent recognition system. Computers & Geosciences, 54, 219-230.

[8] Al Dallal, J. (2015). Identifying refactoring opportunities in object-oriented code: A systematic literature review. Information and software Technology, 58, 231-249.

[9] Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A survey on cyber security for smart grid communications. IEEE communications surveys & tutorials, 14(4), 998-1010.

[10] De Sotomayor, A. A., Della Giustina, D., Massa, G., Dedè, A., Ramos, F., & Barbato, A. (2018). IEC 61850-based adaptive protection system for the MV distribution smart grid. Sustainable Energy, Grids and Networks, 15, 26-33.

[11] Basem, A. M., & Ali, H. (2017). Data Distribution Service (DDS) based implementation of Smart grid devices using ANSI C12. 19 standard. Procedia Computer Science, 110, 394-401.

[12] Ngcobo, T. J., & Ghayoor, F. (2022). An overview of DLMS/COSEM and g3-plc for smart metering applications. International Journal on Smart Sensing and Intelligent Systems, 15(1).

[13] Shanmukesh, P., Mahendra, L., JaganMohan, K., Kumar, R. S., & Bindhumadhava, B. S. (2021). Secure DLMS/COSEM communication for next generation advanced metering infrastructure. Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146, 7(1), 92-98.

[14] Kheaksong, A., & Lee, W. (2014, October). Packet transfer of DLMS/COSEM standards

for smart grid. In The 20th Asia-Pacific Conference on Communication (APCC2014) (pp. 391-396). IEEE.

[15] Vyas, D., & Pandya, H. (2012). Advance metering infrastructure and DLMS/COSEM standards for smart grid. International Journal of Engineering Research &, 1(1).

[16] Tatipatri, N., & Arun, S. L. (2024). A comprehensive review on cyber-attacks in power systems: Impact analysis, detection, and cyber security. IEEE Access, 12, 18147-18167.

[17] Wang, C. L., Shih, J. A., Liao, I. E., & Chien, C. T. (2022, October). An evaluation of cybersecurity risks of dlms/cosem smart meter using fuzzing testing. In 2022 IET International Conference on Engineering Technologies and Applications (IET-ICETA) (pp. 1-2). IEEE.

[18] Jabeen, T., Mehmood, Y., Khan, H., Nasim, M. F., & Naqvi, S. A. A. (2025). Identity Theft and Data Breaches How Stolen Data Circulates on the Dark Web: A Systematic Approach. Spectrum of engineering sciences, 3(1), 143-161.

[19] Mathas, C. M., Vassilakis, C., Kolokotronis, N., Zarakovitis, C. C., & Kourtis, M. A. (2021). On the design of IoT security: Analysis of software vulnerabilities for smart grids. Energies, 14(10), 2818.

[20] Biswas, S., Ghosh, S., Das, P., Saha, K., & De, S. (2023). Efficient Data Transfer Mechanism for DLMS/COSEM Enabled Smart Energy Metering Platform. ACM SIGMETRICS Performance Evaluation Review, 50(4), 14-16.

[21] Ju, S. H., & Seo, H. S. (2018). Design key management system for DLMS/COSEM standardbased smart metering. Int. J. Eng. Technol, 7(3.34).

[22] Chintha, R., & Kumar Chinnaiyan, V. (2018, June). Performance Evaluation of M2M and H2H Communication Coexistence in Shared LTE: A DLMS/COSEM-Based AMI Network Scenario. In Advanced Computational and Communication Paradigms: Proceedings of International Conference on ICACCP 2017, Volume 1 (pp. 177-185). Singapore: Springer Singapore.

[23] Chien, C. T., Wang, C. L., Liao, I. E., & Wang, S. J. (2023, October). Implementing OIML R46 Communication Unit for DLMS/COSEM Security Suite 1 and Passing CTT V3. 1 Test. In 2023 International Conference on Networking, Sensing and Control (ICNSC) (Vol. 1, pp. 1-6). IEEE.

[24] Wülfing, C. A., Reck, F. G., Carloto, F. G., Barriquello, C. H., Marin, P. R., & Nascimento, E. (2022, November). Evaluation of DLMS/COSEM data processing setups applied to smart metering. In 2022 14th Seminar on Power Electronics and Control (SEPOC) (pp. 1-6). IEEE.