Dong *et al. Journal of Cloud Computing* (2024) 13:24
https://doi.org/10.1186/s13677-023-00568-7

Journal of Cloud Computing:
Advances, Systems and Applications

**RESEARCH** **Open Access**

# A new method of dynamic network security analysis based on dynamic uncertain causality graph

Chunling Dong[1], Yu Feng[1] and Wenqian Shang[1*]

**Abstract**

In the context of cloud computing, network attackers usually exhibit complex, dynamic, and diverse behavior characteristics. Existing research methods, such as Bayesian attack graphs, lack evidence correlation and real-time reflection of the network attack events, and high computational complexity for attack analysis. To solve these problems, this study proposes a Dynamic Uncertain Causal Attack Graph (DUCAG) model and a Causal Chain-based Risk Probability Calculation (CCRP) algorithm. The DUCAG model is constructed to represent the uncertain underlying causalities among network attack events, and the CCRP algorithm aims at dynamically updating the causality weights among different network attack events and attacker hypotheses based on alarm information and causal chain reasoning process. By causality simplification and causality reasoning methods, the CCRP efficiently predicts the attacker behaviors and potential attack likelihood under uncertain time-varying attack situations, and is robust to the incompleteness and redundancy in alarm information. Four experiments under different attack scenarios demonstrate that, the DUCAG model can effectively characterize and predict the complex and uncertain attack causalities, in a manner of high time efficiency. The proposed method has application significance on cloud computing platforms by dynamically evaluating network security status, predicting the future behaviors of attackers, and assisting in adjusting network defense strategies.

**Keywords** Dynamic uncertain causality graph, Attack graph, Network attack, Attack scenario construction

## Introduction

In recent years, the rapid development of cloud computing has provided the possibility for the Internet of Things (IoT) to achieve dynamic management and intelligent analysis. Cloud computing has an ability to make the basic IT resources into a resource pool that can be freely scheduled, to realize the on-demand allocation of IT resources, and to provide customers cloud computing services. At present, the cloud computing field is mainly faced with the following problems: resource service scheduling problems, task segmentation problems, network transmission problems, and real-world task application problems. Many research es have been devoted to solving the issues in fields of video distribution [1], traffic management [2], health care [3], intelligent recommendation [4, 5], blockchain[6], privacy-preserving [7] and mobile applications [8, 9]. However, with the development of the IoT, cloud security issues are also becoming more prominent. For example, in 2022, the CSA released the "Top 11 Threats to Cloud Computing", which lists system vulnerabilities, hacker attacks, APT attacks and so on. The SolarWinds incident is also a typical case of APT attacks, affecting more than 300,000 large enterprises worldwide. In the cloud computing environment, a large amount of sensitive data and business applications

*Correspondence:
Wenqian Shang
shangwenqian@cuc.edu.cn
[1] School of Computer and Cyber Sciences, Communication University of China, Beijing 100024, China

are stored and processed, and security has become a research focus in the field.

Cloud security problems are mainly divided into trusted cloud computing problems and protection technology problems. The former mainly includes cloud data security and privacy protection, while the latter refers to the application of traditional security protection technologies and new cloud computing protection technologies for prevention. At present, considerable efforts have been devoted to solving these problems and many effective methods have been proposed. Aiming at data security analysis, song et al. [10] conducted public integrity verification of shared data in the cloud through asynchronous revocation, and the experiment proved that it had high efficiency. Han et al. [11] studied on cloud data integrity audit based on blockchain, providing a relatively comprehensive review of BDIA. Yang et al. [12] proposed ASTREAM anomaly detection method to process abnormal data. Zhang et al. [13] proposed a mechanism for predicting dynamic service deployment based on upcoming stream data, which has been demonstrated to have lower stream processing latency. Xu et al. [14] designed a PW placement method for PWP, which can effectively deal with the placement problem. Regarding privacy protection, Kong et al. [15] proposed a multi-type health data privacy perception prediction method based on local sensitive hashing to achieve a good balance between prediction accuracy and privacy protection. Wang et al. [16] realized privacy-sensing traffic flow prediction in smart cities based on zero-trust multi-sensor data.

Although the above researches have made significant progress in addressing cloud security issues, there are still some problems to be solved. Above all, the modeling and prediction methods for complex and dynamic network security scenarios lack the ability to provide a clear causal interpretability for network analysts. However, low interpretability reduces the credibility and applicability of the methods. Besides, the attacks in the cloud computing environment are characterized by high degree of concealment, uncertainty, dynamics, and complexity, under the attack scenarios with various attack objectives [17]. It has been a challenge to accurately characterize the uncertain causalities among attack events and attacker behaviors. Therefore, it is difficult to utilize the correlations of evidence to construct an accurate attack scenario and reason about the underlying security risk in real time in cloud computing environment. Most of existing researches are based on attack graph modeling methods including attack trees and Bayesian attack graphs. Thus, attentions have less been paid to the correlation among evidence, and high reasoning complexities in these methods reduce the practicality.

To address these problems, this study introduces the theory of Dynamic Uncertain Causality Graph (DUCG) [18–21] into the field of network attack analysis. The methodology of DUCG represents uncertain and complex causalities in a compact fashion of graph model and provides efficient probabilistic reasoning methods for inferences and predictions in complex systems. Based on DUCG, this paper proposes a novel model to effectively characterize the dynamic network attack scenarios and uncertain underlying causalities, and efficiently predict the attacker's capabilities and potential attack likelihood under various attack modes.

The contributions of this study are as follows:

(1) A Dynamic Uncertain Causal Attack Graph (DUCAG) method is proposed for representing and reasoning about the underlying causalities in network attack events. The method not only describes complex and dynamic attack scenarios but also accurately models the characteristics of attacker's attack behaviors in manner of an interpretable causality graph model.

(2) A Causal Chain-based Risk Probability Calculation (CCRP) algorithm is presented to dynamically update and predict the risk probabilities of the nodes in attack event sequences and adjust the relationship weights between different network attack evidence events and attacker hypotheses based on the causal chain reasoning process. Benefitting from the proposed causal chain-based logical reasoning mechanism, CCRP algorithm has high accuracy and computational efficiency for the applications in cloud environment and is robust to the incompleteness and redundancy in alarm information.

The rest of the paper is structured as follows. "Related work" provides an introduction to the related works on modeling methods of network attack scenarios and quantifying the risk probability in attack graph. In "Dynamic Uncertain Causal Attack Graph Method", we describe the principal methodology of DUCAG including the causal graph modeling and risk inference algorithm of CCRP. This is followed by four verification and performance experiments on DARPA dataset and simulated networks in "Experimental Analysis of DUCAG". Finally, we conclude the article in "Conclusions".

## Related work

Existing researches in the field of network attack graph primarily focus on two main directions: the construction of attack scenarios and the quantification of risk associated with each event in the attack scenario. The construction of attack scenarios aims to capture the underlying causalities between various attack behaviors, enabling network administrators to comprehend the current state of network security and establish a foundation for subsequent defense strategies. Quantifying the risk of each event in the attack scenario can assist administrators in evaluating network security status, predicting the future behaviors of attackers, and adjusting defense strategies.

### Researches on the modeling methods of network attack scenarios

For modelling the network attack scenarios and characterizing the causalities related to attack events, Phillips et al. [22] initially proposed the concept of attack graphs, which represent the causal relationships among atomic attacks as directed graphs. The attack graphs are constructed by considering network configuration, attack causality, and other relevant factors. However, state attack graphs are limited by the combinatorial explosion problem of state space. In view of this, subsequent researches have focused on constructing attribute attack graphs to reduce the complexity associated with attack graph modeling [23]. To enable the system administrator to quantify the chances of network compromise at various levels, Poolsappasit et al. [24] constructed Bayesian networks, utilizing the observed state of nodes in the network as evidence for posterior inference. This approach allows for real-time updating of the likelihood of each node being attacked, which is beneficial for the real-time evaluation of network security risks. However, this study only considered the "*AND*" and "*OR*" relationships for depicting the correlations between atomic attacks, and neglected the situation of multiple relationships occurring in real network attacks. Additionally, Bayesian attack graphs are not effective in the absence of alarms. To address the issue of inaccurate attack scenarios caused by missing and redundant alarms, Wang [25] proposed a definition of causal relationships based on expert knowledge and the construction of attack scenarios by applying the causal knowledge network extracted from real alarm data. This method can construct attack scenarios even in the absence of alarms. Furthermore, most previous studies on constructing attack scenarios have focused on identifying single-step attacks. To overcome this limitation, Wang et al. [26] proposed a multi-dimensional correlation analysis of alarm information based on causal knowledge and spatial–temporal associations to construct higher-level attack scenarios, and experiments have shown that the proposed method can effectively depict the complete attack process and reconstruct a high-level attack scenario.

Although a lot of progress have been made on the modeling methods of network attack scenarios, there are still some challenges to be addressed: 1) it is difficult for current methods to accurately represent the time-varying and uncertain scenarios in complex network attack scenarios, as well as to provide effective model interpretability; 2) the modeling method needs to solve the issues of inaccurate attack scenarios related to missing and redundant alarms, while truly reflecting the real-time process of actual network attacks.

### Researches on the quantification of the risk probability in attack graph

Regarding the quantifying of the risk probability associated with each node in the attack graph, Wang [27] introduced probability attributes into the attack graph. However, the calculation of posterior probabilities for nodes in the attack graph when an attack event occurs was not considered. On the basis of Wang's researches, Ye et al. [28] eliminated unreachable paths in the attack graph and proposed to apply the maximum probability of all paths leading to the arrival node to reduce the complexity related to node correlation and loops. The research results demonstrate the effectiveness and rationality of the proposed method. In order to identify the vulnerabilities in the network and their interrelationships, Chen [29] studied the issue of loops in attack graphs by proposing the $n$ valid attack paths representing the real attack process, thereby optimizing the construction of attack graphs. Experiments revealed that the method can effectively reflect the impact of missing data on the assessment results. Regarding the loop issues in attack graph, Ammann [30] proposed the attack monotonicity assumption, which states that once an attacker obtains a certain resource, the attack will not repeat the process of obtaining that resource in subsequent attacks. This assumption resolves the loop problem in attack graphs to some extent. To quantitatively analyze the uncertainties during network attack process, Liu et al. [31] introduced the theory of confidence to represent the credibility of evidence information for hypotheses and applied the DARPA dataset in the experiment, this method can effectively depict the dynamic changes of network security status. Aiming at solving the problem that the attack graph model cannot reflect the real-time network attack events, Li et al. [32] proposed a dynamic risk probability algorithm by integrating the forward and backward updating strategies. The algorithm put forward a method of real-time and accurate evaluation of the risk probabilities for nodes in the attack graph. However, during the calculation process, if a node has already updated its probability value during the forward update process, the node probability will not be updated during the backward updating process. The algorithm overlooks the joint effect of different evidence nodes and hypotheses on the posterior probability of the node.

In summary, for the researches on quantifying risk probability in attack graphs, there are still some unresolved difficult issues: 1) how to accurately predicting the node risk under situations of complex and uncertain casualties associated with both attack event hypothesis and evidence; 2) how to achieve the efficient and dynamic risk assessment in corresponding with the time-varying network attack scenarios.

### Dynamic uncertain causal attack graph method

The fundamental method of DUCG theory introduces independent random events in conjunction with causality graphs to represent and reason about the uncertain events and causalities in complex systems [18–20]. Moreover, efficient causality reasoning algorithms have been proposed for probabilistic inference and prediction. By applying the DUCG method in the domain of network attack analysis, better solutions can be achieved to effectively assess security vulnerabilities, construct network attack scenarios, and predict the future behaviors of attackers. Therefore, this study proposes a Dynamic Uncertain Causal Attack Graph (DUCAG) model based on DUCG, to represent and quantify the uncertain causalities in network attack scenarios. A causal Chain-based Risk Probability Calculation (CCRP) algorithm is proposed to dynamically update the relationship weights among attack events and predict the risk probability of nodes in DUCAG by means of causal chain reasoning process. The mechanisms of uncertain causality representation and inference in DUCG provides a theoretical basis for the proposed DUCAG and CCRP methods under complex situations of network attacks.

"Definition of dynamic uncertain causal attack graph" introduces the definitions of the DUCAG, "DUCAG Simplification and inference algorithms" presents the simplification and hypothetical state inference methods, the definition and implementation steps of the CCRP algorithm are presented in "Algorithm of calculating risk probability based on causality chains", and "Attack scenario construction" proposes the construction method of attack scenarios.

### Definition of dynamic uncertain causal attack graph

As the example of DUCAG illustrated in Fig. 1 (a), an intrusion detection system detects a $V_4$ attack and identifies it as an evidence event. The variable $B_1$ represents a resource node of attacker which implements a specific attack strategy, the $V$-type variable represents an attack behavior, and the $S$-type variable indicates a resource node.

The attack scenarios and characteristics of attacker's attack behaviors are defined as follows.

*Definition 1.* Dynamic Uncertain Causal Attack Graph, DUCAG = ($N$, $E$). The definitions of different variables are shown in Table 1.

### DUCAG Simplification and inference algorithms

When a real-world attack event occurs, the observed attack events can be utilized as evidence to simplify the DUCAG causality graph and calculate the probability of hypothetical states using the DUCAG inference algorithm. The following section presents the causality simplification and inference algorithms of DUCAG.

### Causality simplification and decomposition on the attack causality graph

The causality simplification aims at eliminating the causalities and events that are unconcerned, unreasonable, or inconsistent with the observed evidence from the causality graph [21], thereby reducing the complexity of model and subsequent calculations. For the cases that the causality graph is large in scale and contains multiple independent root cause event hypotheses, it is necessary to decompose the attack causality graph into multiple sub-DUCAG based on each of the root event hypotheses.

Figure 2 illustrates the process of constructing an attack scenario by utilizing DUCAG. Based on the attack scenarios model as represented by the original DUCAG

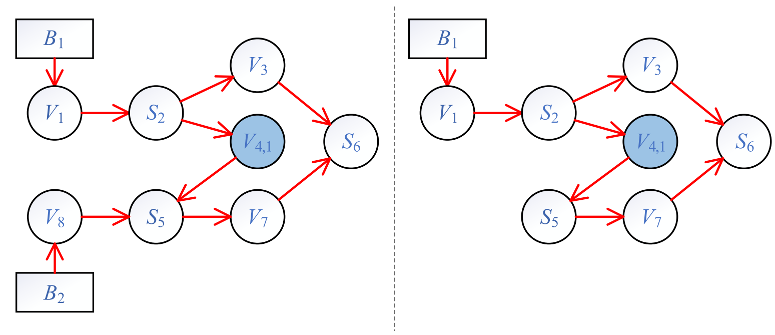

**Fig. 1** **a** Example of a DUCAG, (**b**) The simplified DUCAG of the $B_1$ hypothesis

**Table 1** Definition and description of each type of variable

| Variable | Description |
|---|---|
| $X_n$ | *X*-type variable represents an observable alarm data. The variable number is denoted by "*n*". In the attack graph, an *X*-type variables contain two forms: the resource node represented by the symbol *S*, and the attack behavior node represented by the symbol *V*. |
| $B_i$ | *B*-type variable represents the root cause event hypothesis, and the subscript "*i*" is the variable number. The attack assumptions serve as a useful way to assess the attacker's capabilities and objectives in carrying out the attack. |
| $G_k$ | *G*-type variable represents the logic gate variable, and the subscript "*k*" is the variable number. In the attack graph, logic gate variables can represent complex logical relationships between parent nodes and child nodes, such as "*AND*", "*OR*", "*XOR*", and so on. |
| $r_{n;i}$ | "$r_{n;i}$" denotes the degree of causal association between a parent node "*i*" and a child node "*n*". This measure serves to quantify the causal influence exerted by the parent node on the child node. |
| → | The weighted function variable is a directed edge in the attack graph, denoted as $F_{n;i} \equiv (r_{n;i}/r_n)A_{n;i}$. It indicates the causal function between the parent variable $X_i$ and the child variable $X_n$. There are two types of weighted function variables: one represents the probability of transitioning from a state node to an attack behavior, and the other represents the probability of transitioning from an attack behavior to a state node. |
| $A_{n,k;i,j}$ | $A_{n,k;i,j}$ represents the uncertain function mechanism of the $X_i$ in state *j* independently causing the variable $X_n$ in state *k*. Probability transition matrices quantify the uncertainty of causal functions among variables. The parameters can be obtained from statistical learning or domain knowledge. |

in Fig. 2 (a), sequential abnormal alarms may be observed at different times. The observed alarm evidence is represented as blue-colored node in the graph, as shown in the Fig. 2 (b). By performing causality simplification, the events that are irrelevant with the observed evidence and concerned hypothesis are removed from DUCAG to get a simplified attack causality graph. Thus, based on an original DUCAG in terms of different root event hypotheses and real-time alarm information, different simplified attack causality graphs can be obtained as illustrated in Fig. 2 (b).

### Hypothesis event state probability calculation

The calculation of the posterior probability of the Hypothesis event can be performed by using the weighted logical inference method proposed in [21]. This method allows for the determination of the posterior probability of the specific hypothesis event, Pr ($H_{i,j}$ $|E$), where $H_{i,j}$ represents the root event hypothesis. The weighted logical inference method has two steps: logical reasoning and probabilistic reasoning. The logical reasoning involves expanding all evidence nodes along the causality chains, up forward to the initial cause events $B_i$. The expansion of node $X_{n,k}$ at the event level can be accomplished by formula (1):

$$X_{n,k} = \sum_{p} \left(r_{n;p}\middle/r_n\right) \sum_{q} A_{n,k;p,q}\rho_{p,q} \tag{1}$$

In which $\left(r_{n;p}\middle/r_n\right)$ quantifies the causal influence of different parent nodes on node $X_{n,k}$, $r_n = \sum_p r_{n;p}$. $\rho_{p,q}$ represents parent node $\rho$ in state $q$. $A_{n,k;p,q}$ represents the causal function mechanism of $\rho_{p,q}$ causing $X_{n,k}$. Based on the causality expressions of each evidence, the weighted logic "*AND*" operations are performed on all expressions, and the conditional state probability of the hypothesis event $H_{i,j}$ can be obtained by formula (2):

$$h^s_{i,j} \equiv \Pr\{H_{i,j}|E\} = \frac{Pr\{H_{i,j}E\}}{Pr\{E\}} = \frac{\Pr\{H_{i,j} \cap X_{p,q}\}}{\Pr\{\cap X_{p,q}\}} \tag{2}$$

The probabilistic reasoning process is to replace each variable in the expression with the corresponding prior probability value, and after algebraic operations and normalization processing, the conditional posterior probability of the hypothesis can be obtained.

### Algorithm of calculating risk probability based on causality chains

To dynamically adjust the relationship weights between different network attack evidence events and attacker hypotheses based on alert information and causal chain reasoning process, a Causal Chain-based Risk Probability Calculation (CCRP) algorithm is proposed in this study. The posterior probability of a node is determined by both the root event hypothesis and the evidence event (attack event). The CCRP algorithm has three steps:

(1) Expansion of the causal chain of nodes;
(2) Elimination of invalid causal chains;
(3) Calculation of the posterior probability based on the causal chains.

### The causal chain-based expansion algorithm

The algorithm for calculating posterior probabilities for general causal variables involves two steps:

(1) Logical reasoning.

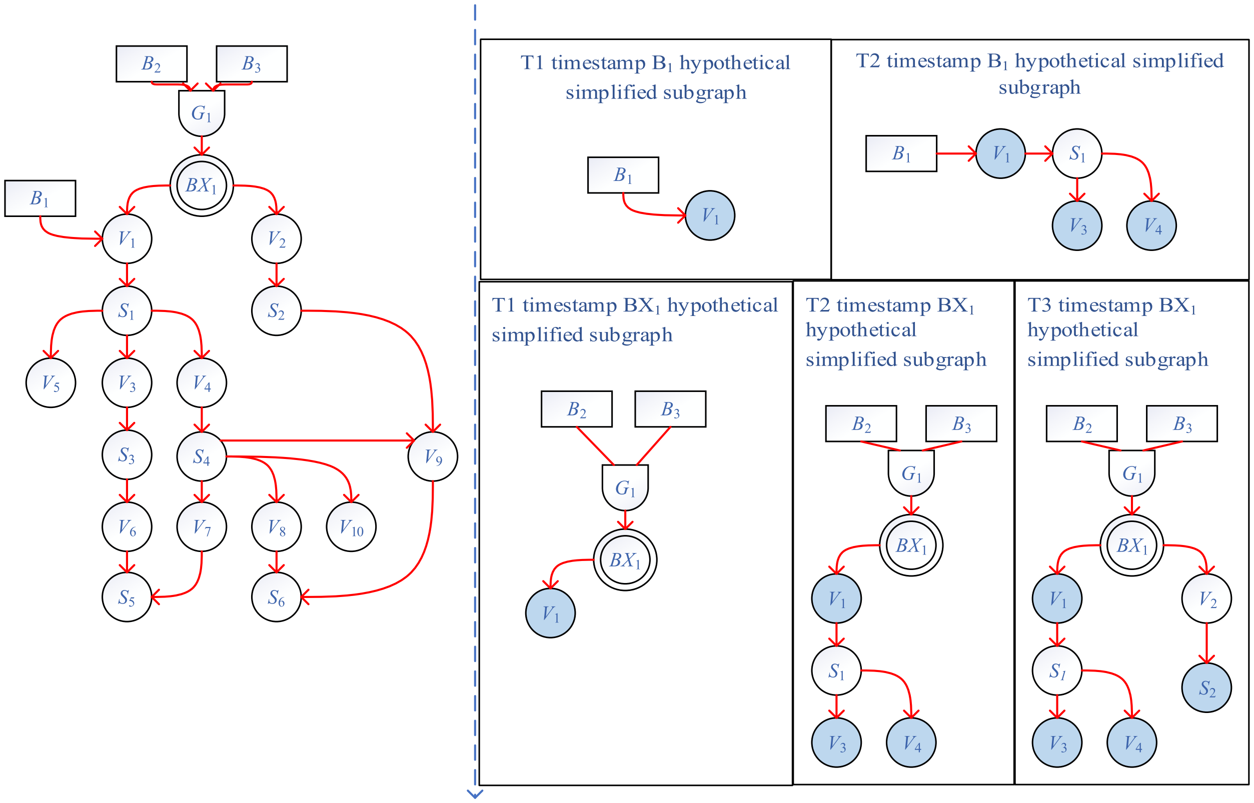
$B_2$
$B_3$
$G_1$
$BX_1$
$B_1$
$V_1$
$V_2$
$S_1$
$S_2$
$V_5$
$V_3$
$V_4$
$V_9$
$S_3$
$S_4$
$V_6$
$V_7$
$V_8$
$V_{10}$
$S_5$
$S_6$
T1 timestamp $B_1$ hypothetical simplified subgraph
T2 timestamp $B_1$ hypothetical simplified subgraph
T1 timestamp $BX_1$ hypothetical simplified subgraph
T2 timestamp $BX_1$ hypothetical simplified subgraph
T3 timestamp $BX_1$ hypothetical simplified subgraph

**Fig. 2** **a** An example of DUCAG, (**b**) The causality simplification of the DUCAG

Note that at time T3, the $B_1$ hypothesis is no longer able to explain all the observed alarm evidence, so the hypothesis $B_1$ and its simplified graph is removed and only the $BX_1$ hypothesis with its simplified graph is reserved

Logical reasoning needs to expand all relevant event expressions of a concerned variable upward and backward, until encountering a root cause node, terminal node, or evidence node. There are three types of causality chains in the process of expanding the node's causal chains: Unknown Causal Chains, Evidence Causal Chains, and Hypothesis Causal Chains.

The Unknown Causal Chain (UCC) denotes the causal chains in which evidence events or hypothesis events are not included in the event expansion process. All the UCCs of node $X_v$ can be represented as $\sum \mathrm{UCC}(X_v)$. The Evidence Causal Chain (ECC) refers to causal chains that include evidence events. All the ECCs of node $X_v$ can be represented as $\sum \mathrm{ECC}(X_v)$. The Hypothesis Causal Chain (HCC) refers to the causal chains that include hypothesis events. All the HCCs of node $X_v$ can be represented as $\sum \mathrm{HCC}(X_v)$. All the causal chains associated with a node are denoted by $\mathrm{Causal\text{-}Chains}(X_v)$ in the DUCAG, and formula (3) always holds in the causal network.

(2) Probabilistic reasoning.

Probabilistic reasoning involves substituting the prior probabilities of the nodes into the causal chain for probability calculation.

### Elimination of invalid causal chains

To calculate the posterior probability of a node under the scenario of an attack event, it is necessary to eliminate the invalid causal chains. The term "invalid causal chains" refers to the UCC of a concerned node. In practical network attacks, UCCs do not provide meaningful information about the node and are considered redundant in the calculation process. Therefore, the UCCs are eliminated before reasoning process. The causal chain of node $X_v$ after removing the unknown causal chains can be denoted as $\mathrm{Causal\text{-}Chains}(X_v)*$, and formula (4) holds as follows:

$$\mathrm{Causal\text{-}Chains}(X_v) = \sum \mathrm{UCC}(X_v) + \sum \mathrm{ECC}(X_v) + \sum \mathrm{HCC}(X_v) \tag{3}$$

$$\text{Causal - Chains}(X_v)^* = \sum \text{ECC}(X_v) + \sum \text{HCC}(X_v) \quad (4)$$

### Posterior probability calculation based on causal chains

The calculation of posterior probability based on causal chains refers to the computation of the results by substituting the corresponding parameters with all connected events or hypothetical events in the causal chain of node $X_v$ after eliminating ineffective causal chains. The posterior probability of the node can be calculated using formula (5).

$$\Pr(X_v) = \frac{\text{Causal - Chains}(X_v)^*}{N} = \frac{\sum \text{ECC}(X_v) + \sum \text{HCC}(X_v)}{N} \quad (5)$$

In formula (5), the value of $N$ represents the sum of all distinct evidence nodes and hypotheses in the causal chains associated with the current node. By performing logical reasoning on the causal chains of evidence and hypotheses, then dividing the sum by the number of distinct evidence and hypotheses, the algorithm automatically assigns different weights to different causal chains. This process dynamically adjusts the contribution of calculating the risk probability between evidence nodes, and between evidence and hypotheses. If a node has multiple evidence chains associated with different evidence events, the higher weights are assigned while the weights of the hypothesis chains will be reduced.

### Attack scenario construction

The attack scenario is a description of a series of attack behaviors. In a DUCAG, the attack scenario can be formalized as follows:

$$Attack = \{B_i \rightarrow S_1 \rightarrow V_1 \rightarrow \ldots \rightarrow S_n \rightarrow V_n\}$$

The key to constructing a reasonable attack scenario is to combine related causal networks to find the most probable attack sequence by means of causality inference methods. From a qualitative perspective, it refers to simplifying the causal network based on the causality simplification methods after detecting the attack events, to simplify the complexity of the causality model. Further, a weighted logical reasoning algorithm is presented to obtain the posterior probability of each node in the attack graph, and based on the posterior probability of the nodes, the posterior probability of different attack sequences is calculated to determine the attack sequence with the maximum posterior probability as the final constructed attack scenario. In summary, the reconstruction of attacks based on DUCAG has four steps.

*Step 1.* In the process of constructing attack scenario, the causality inference process utilizes the node sequence formed by the alarm information at time $T_i$ as evidence variables to simplify the causal graph, and the algorithm of CCRP is performed to calculate the posterior probability of each node in the causal network.

*Step 2.* Calculate the posterior probability of different attack sequences from the source node to the target node in the causal network. The calculation method is shown in formula (6).

$$\Pr(AttackSeq_i) = \Pr(B_i) \times \Pr(S_1) \times \Pr(V_1) \times \ldots \times \Pr(S_k) \times \Pr(V_k) \quad (6)$$

*Step 3.* Scale the posterior probabilities of different attack path sequences. In a DUCAG, multiple hypotheses may have the same attack path (excluding hypothesis events). If an attack path sequence can be valid in different hypotheses, its posterior probability needs to be adjusted to fit the correct real situation. Therefore, all the posterior probability of attack path sequence of the different hypotheses are introduced to depict the attack scenarios. The calculation method of the adjusted posterior probabilities of attack path sequences is shown in formula (7), where $\{B\}$ represents the hypothesis space.

$$\Pr(AttackSeq_i)* = \sum_{i,k}^{\{B\}} \Pr(B_i) \times \Pr(S_1) \times \Pr(V_1) \times \ldots \times \Pr(S_k) \times \Pr(V_k) \quad (7)$$

*Step 4.* Normalize the posterior probabilities of different attack sequences, and then select the attack sequence with the maximum posterior probability as the final constructed attack scenario.

A calculation example (the graph is shown in Fig. 1, and the parameters are listed in Table 2) is provided below to

**Table 2** Parameters related to attack graph instances

| Attack Path | Transition Probability | Attack Path | Transition Probability |
|---|---|---|---|
| $B_1 \rightarrow V_1$ | 0.845 | $S_2 \rightarrow V_3$ | 0.284 |
| $V_1 \rightarrow S_2$ | 1 | $S_2 \rightarrow V_4$ | 0.572 |
| $V_4 \rightarrow S_5$ | 1 | $S_7 \rightarrow V_6$ | 1 |
| $V_3 \rightarrow S_6$ | 1 | $S_5 \rightarrow V_7$ | 0.238 |
| $B_2 \rightarrow V_8$ | 0.155 | $V_8 \rightarrow S_5$ | 1 |

illustrate the above steps. For the sake of simplicity, the causal effect weights in this example are set to 1, i.e., $r=1$.

As shown in Fig. 1 (b), the blue node represents this evidence variable, and the node state is set to 1. we have confirmed that the attacker will use the $B_1$ strategy, so we set the posterior probability of $B_1$ to 1. $B_1 \rightarrow V_1$ indicates that there is 0.845 possibility that the attacker will launch a $V_1$ attack, and $V_1 \rightarrow S_2$ indicates that there is a 100% possibility of successfully obtaining the resources of node $S_2$ after launching a $V_1$ attack. The posterior probability of each node is calculated based on the evidence and hypothesis-based posterior probability algorithm proposed above. The calculation process is shown in Table 3.

Based on the observed evidence representing attack events, the posterior probabilities of all nodes are calculated. Then, the attack scenarios can be constructed from the source nodes to the target nodes. Taking $B_1$ to $S_6$ as an example, $B_1$ node is the attack source point, $S_6$ is the target node, and there are two attack paths between the two nodes: $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_6$ and $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_7 \rightarrow S_6$. Applying the (7), the posterior probabilities of the two paths are calculated respectively:

$$\Pr(B_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_6) : 1{*}0.7085{*}0.7085{*}0.23998{*}0.23998 = 0.0289$$

$$\Pr(B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_6) : 1{*}0.7085{*}0.7085{*}1{*}1{*}0.238{*}0.23998 = 0.02867$$

After normalization processing, the following results can be got:

$$\Pr(B_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_6) = 0.5019,\ \Pr(B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow V_7 \rightarrow S_6) = 0.4981$$

## Experimental analysis of DUCAG

To validate the effectiveness of DUCAG, four experiments are carried out. Experiment 1 is performed according to the experimental approach in Ref [33], by utilizing the DARPA2000 dataset provided by the Lincoln Laboratory and the causal network extracted from the dataset in Ref [25]. The experimental results of the DUCAG model are qualitatively analyzed in the context of LLDOS attack scenarios. Experiment 2 is according to the method presented in Ref [25] which constructed a small-scale simulated attack-defense environment. Experiment 3 is based on the simulated attack-defense scenario described in Ref [35] which aims to measure the dynamic risk probabilities of various nodes in the network. Experiment 4 compares the time complexity of the CCRP algorithm and the Bayesian inference algorithm based on different scale networks and evidences.

**Table 3** The process of calculating the posterior probability of attack graph nodes

| Node | Causal Chain Expression | Posterior Probability |
|---|---|---|
| $B_1$ | - | 1 |
| $V_1$ | $1/2(A_{1;1}+A_{2;1}{*}A_{4,1;2})$ | 1/2(0.845*1+0.572)=0.7085 |
| $S_2$ | $1/2(A_{1;1}A_{2;1}+A_{4,1;2})$ | 1/2(0.845*1+0.572)=0.7085 |
| $V_3$ | $A_{1;1}{*}A_{2;1}{*}A_{3;2}$ | 0.845*1*0.284=0.23998 |
| $V_{4,1}$ | - | 1 |
| $S_5$ | $A_{5;4,1}$ | 1*1=1 |
| $S_6$ | $1/2{*}\ A_{1;1}\ {*}\ A_{2;1}\ {*}\ A_{3;2}\ {*}\ A_{6;3}+$<br>$1/2{*}\ A_{5;4,1}\ {*}\ A_{7;5}\ {*}\ A_{6;7}$ | 1/2*(0.845*1*0.284*1+0.238)<br>=0.23899 |
| $V_7$ | $A_{5;4,1}{*}A_{7;5}$ | 1*0.238=0.238 |

### Experiment 1: attack scenario construction based on DARPA dataset

This experiment is performed based on the DARPA dataset which has been released by MIT Laboratory and is commonly used for constructing network attack scenarios. The complete attack sequence in current DARPA dataset mainly consists of five stages [34]:

(1) During the initial reconnaissance phase, the attacker employs a script to scan various addresses using IP Sweep. Simultaneously, they listen for ICMP responses to determine which hosts can be targeted for intrusion.

(2) In the subsequent reconnaissance phase, the attacker utilizes Sadmind Ping to scan for the Sadmind daemon service port, identifying which hosts are running the Sadmind program.

(3) In the first stage of intrusion, the attacker exploits vulnerabilities in the Sadmind tool to gain unauthorized access to the system, attempting to execute commands remotely with root privileges.

(4) Once the attacker successfully obtains root privileges, they proceed to install the Mstream Trojan on the victim's host, which can be used for DDoS attacks.

(5) In the final stage of intrusion, the attacker gains access to the victim's host via telnet and utilizes the compromised host to launch DDoS attacks.

A DUCG causality graph is developed for representing the causalities in LLDOS1.0 attack process to model and reason about the attack scenarios. The construction process of the DUCG causality graph is as follows:

(1) Determine the {$X$ ($S$, $V$), $B$}-type variables in the DARPA system. The defined variables are listed in Table 4. The original attack node in the attack sequence is identified as a $B$-type variable. There are four nodes defined as original attack nodes in Table 4. The resource variable $S$ and the atomic attack variable $V$ are defined as the $X$-type variable.

(2) The {$X$, $B$} type variables are divided into several states. The resource variables and atomic attack variables in the $X$ variable have only two states. The resource variable has two states: a normal state denoted by *TRUE*, and an abnormal state denoted by *FALSE*. The atomic attack variable has two states: the attack launched denoted by *FALSE*, and the attack not launched denoted by *TRUE*.

(3) The probability parameters for each state of the $B$-type variables are determined. In the LLDOS attack process, there are four $B$-type variables. Taking variable $B_1$ as an example, $B_1$ has two states: attack not launched representing the probability of not using this operation method, and attack launched representing the probability of performing a DNS Query operation.

(4) For each $X$-type variable, a DUCG subgraph is constructed by the following steps:

1) Select a variable of type atomic attack ($V$) as the module variable in the $X$-type variables;
2) Determine the parent variable of the $X$ variable from the already defined {$X$, $B$}-type variables;
3) Connect the parent variable to the child variable $X_n$ using action variables or conditional action variables; the logical gate variable $G$ can be introduced to represent the complex logical relationships among variables;
4) Determine the association degree $r$ and transition probability matrix between each action variable and conditional action variable (Refer to [25, 31] for the parameters in transition probability matrix);
5) Merge the subgraphs to create a complete DUCG causality graph.

**Table 4** Variables in DUCAG

| **Variable** | **Description** |
|---|---|
| $B_2$ | IP sweep operation |
| $X_2, X_{10}, X_{17}, X_{24}, X_{42}, X_{12}, X_{13}, X_{28}, X_{35}, X_{33}$ | Sadmind Ping operation |
| $X_{30}, X_{15}, X_{14}, X_{27}, X_{41}, X_{36}$ | Sadmind Ping Exploit operation |
| $X_6, X_{32}, X_{44}$ | Daemon Installed operation |
| $X_8, X_{34}, X_{46}$ | DDoS attack |
| $B_1, B_3, B_4$ | DNS Query operation |
| $X_1, X_3, X_4, X_{26}, X_{40}, X_5, X_7, X_9, X_{11}, X_{19}, X_{20}, X_{21}, X_{22}, X_{3,}X_{29}, X_{35}, X_{37}, X_{28}, X_{39}, X_{43}, X_{46}, X_{16}, X_{18}, X_{25}$ | Sources |
| $G_1$ | $X_3 \cup X_{11}$ |
| $G_2$ | $X_{11} \cup X_{18}$ |
| $G_3$ | $X_{11} \cup X_{25}$ |

By performing the above steps, a DUCAG of the LLDOS attack process can be constructed, as shown in Fig. 3.

The experiment process is as follows:

(1) Initialize simulated attack sequences and alarm sequences;

(2) Perform the DUCAG reasoning process to calculate the posterior probabilities of each node in the attack graph;

(3) Identify the node sequence with the highest posterior probability, and compare it with the simulated attack sequence to determine whether the simulation effectively represents the true attack scenarios.

Table 5 shows the specific details of the experiment configuration, where the first simulated attack sequence is a partial attack process of LLDOS1.0, and the second path simulates a partial attack alarm of LLDOS2.0.

The simplified DUCAG graphs for Path One and Path Two are shown in Fig. 4

In Path One, it can be seen from the graph that only the hypothesis of $B_2$ holds after receiving evidence such as $X_4$, $X_{11}$, $X_{26}$, and $X_{40}$. Thus, we obtain the attack scenario through qualitative methods. The attack scenario is $B_2 \rightarrow X_9 \rightarrow X_{10} \rightarrow X_{11} \rightarrow X_4$ ($X_{26}$, $X_{40}$).

In Path Two, we can note that there exists a path: $B_3 \rightarrow X_{16} \rightarrow X_{17} \rightarrow X_{18} \rightarrow X_{26} \rightarrow X_{31} \rightarrow X_{32} \rightarrow X_{37} \rightarrow X_{34} \rightarrow X_{29}$, which has the maximum posterior probability under current evidence. Other paths involve more unconfirmed nodes than this one. The attacked scenes are obtained by applying causality simplification methods and the results prove the validity of constructing attacked scenes using DUCAG.

### Experiment 2: attack scenario construction based on simulated networks

Figure 5 shows the causal network constructed in Ref [25] and its corresponding DUCAG, respectively. The causal

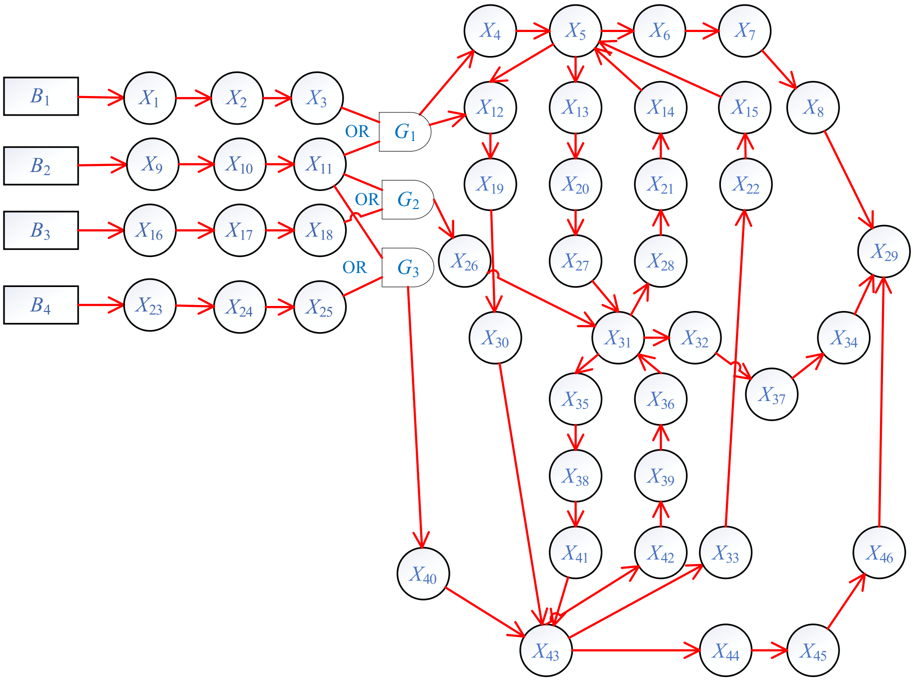

**Fig. 3** LLDOS DUCAG

**Table 5** Configuration of experiment 1

| Simulate Attack Sequence | Attack Alert Sequence | Candidate Sequence |
|---|---|---|
| $B_1$, $X_9$, $X_{10}$, $X_{11}$, $X_4$, $X_5$, $X_6$, $X_7$, $X_8$, $X_{26}$, $X_{31}$, $X_{32}$, $X_{37}$, $X_{34}$, $X_{29}$, $X_{40}$, $X_{43}$, $X_{44}$, $X_{45}$, $X_{46}$ | $X_{11}$, $X_{26}$, $X_{40}$, $X_4$(Missing: $B_2$, $X_9$, $X_{11}$, $X_{26}$, $X_{32}$, $X_{34}$, $X_{29}$, $X_4$, $X_6$, $X_7$, $X_8$, $X_{29}$, $X_{40}$, $X_{44}$, $X_{45}$) | $B_2 \rightarrow X_9 \rightarrow X_{10} \rightarrow X_{11} \rightarrow X_4$ ($X_{26}$, $X_{40}$) |
| $B_2$, $X_{16}$, $X_{17}$, $X_{18}$, $X_{26}$, $X_{31}$, $X_{32}$, $X_{37}$, $X_{34}$, $X_{29}$, $X_{36}$, $X_{39}$, $X_{42}$, $X_{43}$, $X_{44}$, $X_{45}$, $X_{46}$ | $X_{17}$, $X_{18}$, $X_{31}$, $X_{32}$, $X_{37}$, $X_{34}$, $X_{29}$, $X_{36}$, $X_{39}$, $X_{42}$, $X_{43}$, $X_{46}$ (Missing: $X_{16}$, $X_{44}$, $X_{45}$) | $B_3 \rightarrow X_{16} \rightarrow X_{17} \rightarrow X_{18} \rightarrow X_{26} \rightarrow X_{31} \rightarrow X_{32} \rightarrow X_{37} \rightarrow X_{34} \rightarrow X_{29}$ |

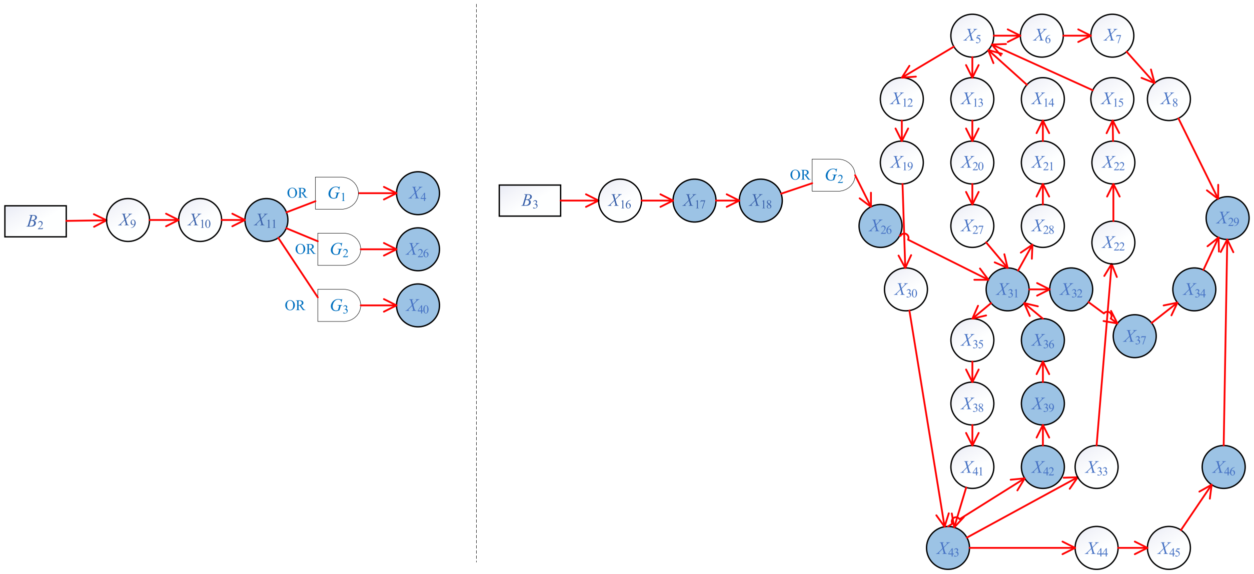

**Fig. 4** **a** The simplified DUCAG graph for Path One, (**b**) The simplified DUCAG graph for Path Two

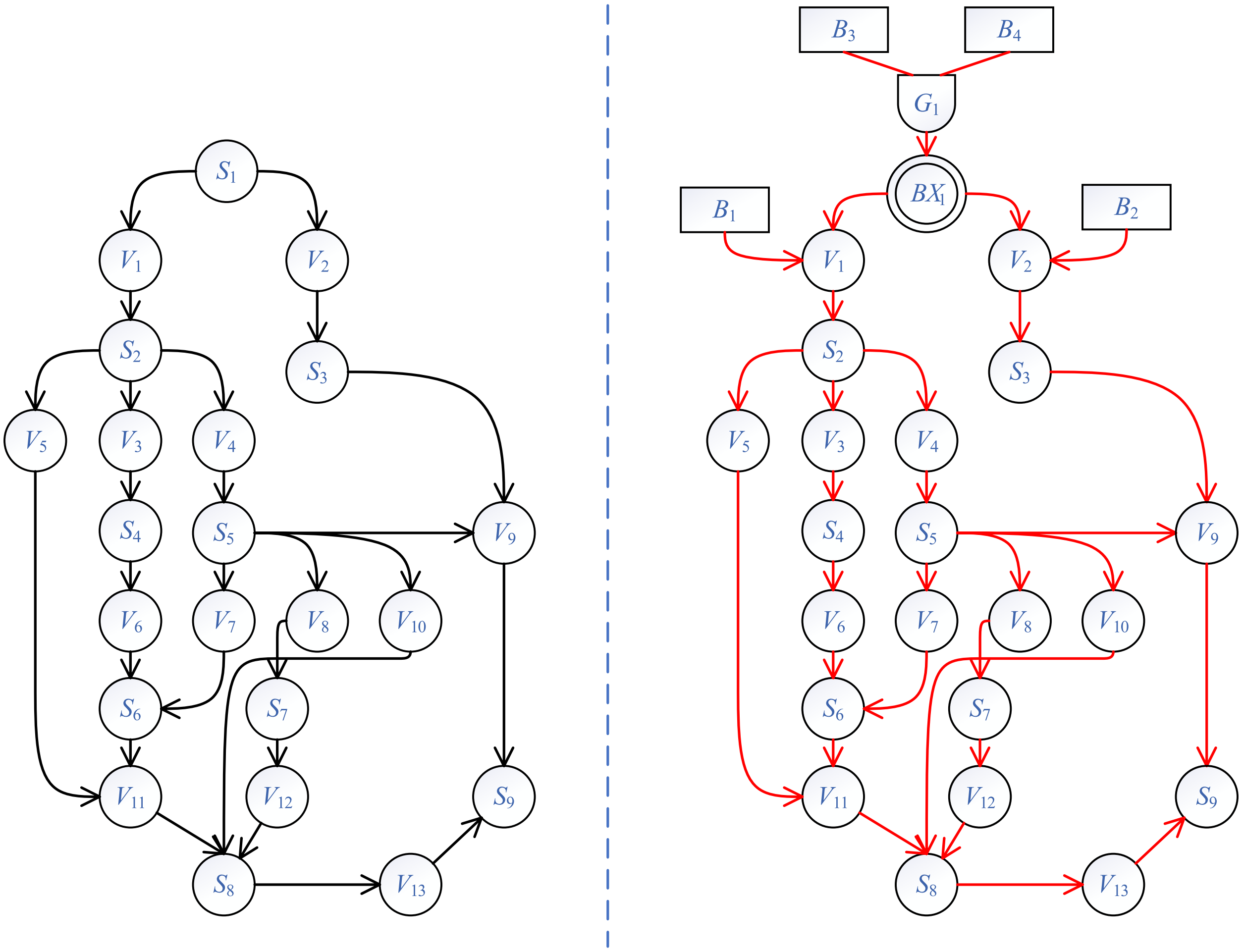

**Fig. 5** **a** The causal knowledge network structure in Ref [25], **b** The corresponding DUCAG graph

parameters obtained through statistical significance testing in Ref [25] are utilized in this study. We introduce the integrated variable to represent an attacker's behavior of adopting multiple attack strategies in the initial attack phase. For instance, in this example the integrated variable $BX_1$ represents the situation that the attacker simultaneously applies both $v_1$ and $v_2$ attack strategies.

Table 6 lists the transition probabilities between different nodes. In this experiment, the logic gate represents a "*AND*" relationship which indicates that attacker take multiple strategies to attack this network.

This study refers to the approach in [25] for simulating the three different attack paths, to verify the effectiveness of DUCAG for dynamic risk calculation. The three simulated attack paths are:

(1) Path 1: $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_4 \rightarrow V_6 \rightarrow S_6 \rightarrow V_{11} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$.
(2) Path 2: $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_5 \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$.

**Table 6** Each attack path in a causal network and its probability of transfer

| Attack Path | Transition Probability | Attack Path | Transition Probability |
|---|---|---|---|
| $BX_1 \rightarrow V_1$ | 1 | $BX_1 \rightarrow V_2$ | 1 |
| $G_1 \rightarrow BX_1$ | 0.5 | $B_2 \rightarrow V_2$ | 0.155 |
| $B_1 \rightarrow V_1$ | 0.845 | $S_2 \rightarrow V_3$ | 0.284 |
| $V_2 \rightarrow S_3$ | 1 | $S_2 \rightarrow V_5$ | 0.144 |
| $V_3 \rightarrow S_4$ | 1 | $S_3 \rightarrow V_9$ | 1 |
| $V_4 \rightarrow S_5$ | 1 | $S_4 \rightarrow V_6$ | 1 |
| $V_5 \rightarrow S_8$ | 1 | $S_5 \rightarrow V_7$ | 0.238 |
| $V_6 \rightarrow S_6$ | 1 | $S_5 \rightarrow V_8$ | 0.301 |
| $V_7 \rightarrow S_6$ | 1 | $S_5 \rightarrow V_9$ | 0.286 |
| $V_8 \rightarrow S_7$ | 1 | $S_5 \rightarrow V_{10}$ | 0.175 |
| $V_9 \rightarrow S_9$ | 1 | $S_6 \rightarrow V_{11}$ | 1 |
| $V_{10} \rightarrow S_8$ | 1 | $S_7 \rightarrow V_{12}$ | 1 |
| $V_{13} \rightarrow S_9$ | 1 | $S_8 \rightarrow V_{13}$ | 1 |

(3) Path 3: $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_8 \rightarrow S_7 \rightarrow V_{12} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$.

Next, the DUCG causality simplification rules are applied to simplify the causality graphs of different simulated attack paths. Figure 6 shows the DUCAG corresponding to the simulated path 1.

Table 7 lists the configuration of experiment 2. Then the CCRP algorithm is applied to calculate the posterior probabilities of nodes related to simulated attack path 1. The calculation process is listed in Table 8, and the calculation process for $BX_1$ is not listed here limited to the space. Note that some alarms are missing or redundant. Benefitting from the above proposed weighted

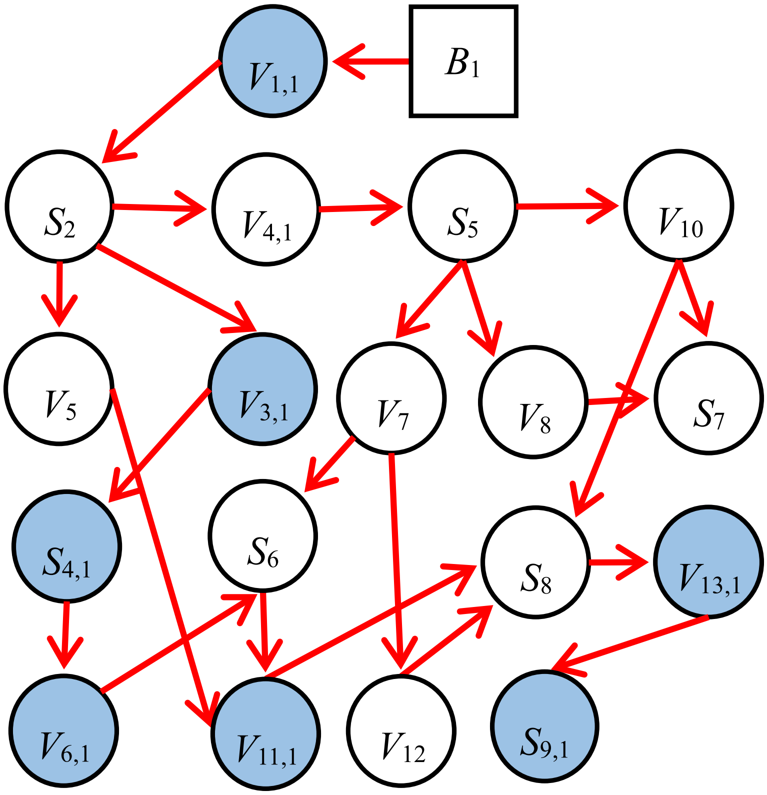

**Fig. 6** The DUCAG corresponding to the simulated path 1

**Table 7** Different simulated paths and the experimental results

| Simulation Path | Alarm Sequence | Best Candidate Sequence and Its Probability | Candidate Paths in [25] |
|---|---|---|---|
| $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_4 \rightarrow V_6 \rightarrow S_6 \rightarrow V_{11} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$ | $V_1 \rightarrow V_3 \rightarrow S_4 \rightarrow V_6 \rightarrow V_{11} \rightarrow V_{13} \rightarrow S_9$<br>(Missing alarms: $S_2$, $S_6$, $S_8$) | $B_1(BX_1) \rightarrow V_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_4 \rightarrow V_6 \rightarrow S_6 \rightarrow V_{11} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$(0.3542)<br>$BX_1 \rightarrow V_2 \rightarrow S_3 \rightarrow V_9 \rightarrow S_9$(0.2941)<br>$B_1(BX_1) \rightarrow V_1 \rightarrow S_2 \rightarrow V_5 \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$(0.1293) | $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_4 \rightarrow V_6 \rightarrow S_6 \rightarrow V_{11} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$<br>$B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_7 \rightarrow S_6 \rightarrow V_{11} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$ |
| $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_5 \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$ | $V_1 \rightarrow S_2 \rightarrow V_5 \rightarrow V_3 \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$<br>(Redundant alarms: $V_3$) | $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_5 \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$(0.8839)<br>$BX_1 \rightarrow V_2 \rightarrow S_3 \rightarrow V_9 \rightarrow S_9$(0.0622) | $S_5 \rightarrow V_8 \rightarrow S_7 \rightarrow V_{12} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$<br>$B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_3 \rightarrow S_4 \rightarrow V_6 \rightarrow S_6 \rightarrow V_{11} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$ |
| $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_8 \rightarrow S_7 \rightarrow V_{12} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$ | $V_1 \rightarrow S_2 \rightarrow S_5 \rightarrow V_9 \rightarrow V_8 \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$<br>(Redundant alarms: $V_9$, Missing alarms: $V_4$, $S_7$, $V_{12}$) | $B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_8 \rightarrow S_7 \rightarrow V_{12} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$(1882)<br>$B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_5 \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$(0.1207)<br>$BX_1 \rightarrow V_2 \rightarrow S_3 \rightarrow V_9 \rightarrow S_9$(0.3449)<br>$BX_1(B_1) \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_9 \rightarrow S_9$(0.2409) | $B_1 \rightarrow V_2 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_8 \rightarrow S_7 \rightarrow V_{12} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$<br>$B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_7 \rightarrow S_6 \rightarrow V_{11} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$<br>$B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_{10} \rightarrow S_8 \rightarrow V_{13} \rightarrow S_9$<br>$B_1 \rightarrow V_1 \rightarrow S_2 \rightarrow V_4 \rightarrow S_5 \rightarrow V_2 \rightarrow S_3 \rightarrow V_9 \rightarrow S_9$ |

**Table 8** Assuming the scenario $B_1$, the calculation process for the posterior probability of nodes in Path 1

| Node | Causal Chain Expression | Posterior Probability |
|---|---|---|
| $V_1$ | - | 1 |
| $S_2$ | $1/4*(A_{2;1,1}+A_{4,1;2}+A_{3,1;2}+1/4*A_{5;2}A_{8;5}A_{13,1;5})$ | 0.473 |
| $V_5$ | $1/2*(A_{2;1,1}A_{5,2}+1/4*A_{8;5}A_{13,1;8})$ | 0.197 |
| $V_3$ | - | 1 |
| $V_4$ | - | 1 |
| $S_4$ | - | 1 |
| $S_5$ | $1/3*(A_{5;4}+1/2*A_{7;5}A_{6;7}A_{11,1;6}+1/4*A_{8;5}A_{7;8}A_{12;7}A_{8;12}A_{13;8}+1/4*A_{10;5}A_{8;10}A_{13,1;8})$ | 0.4126 |
| $V_6$ | - | 1 |
| $V_7$ | $1/2*(A_{5;4}A_{7;5}+1/2*A_{11,1;6}A_{6;7})$ | 0.369 |
| $V_8$ | $1/2*(A_{5;4}A_{8;5}+1/4*A_{13,1;8}A_{8;12}A_{12;7}A_{7;8})$ | 0.2755 |
| $V_{10}$ | $1/2*(A_{5;4}A_{10;5}+1/4*A_{8;10}A_{13,1;8})$ | 0.2125 |
| $S_6$ | $1/3*(1/2*A_{5;4}A_{7;5}A_{6;7}+1/2*A_{6,6}+A_{11,1;6})$ | 0.5396 |
| $S_7$ | $1/2*(A_{5;4}A_{8;5}\ A_{7;8}+1/4*A_{13,1;8}A_{8;12}A_{12;7})$ | 0.2755 |
| $V_{11}$ | - | 1 |
| $V_{12}$ | $1/2*(A_{5;4}A_{8;5}A_{7;8}A_{12;7}+1/4*A_{13,1;8}A_{8;12})$ | 0.2755 |
| $S_8$ | $1/4*(A_{13,1;8}+1/4*(A_{2;1,1}A_{5,2}A_{8;5}+A_{8;11}+A_{5;4}A_{8;5}A_{7;8}A_{12;7}A_{8;12}+A_{5;4}A_{10;5}A_{8;10})$ | 3512 |
| $V_{13}$ | - | 1 |
| $S_9$ | - | 1 |

logical causality reasoning mechanism, CCRP algorithm is robust to the incompleteness and redundancy in observed alarm information.

Taking simulation path 1 as an example, the evidence set observed is $\{V_1, V_3, S_4, V_6, V_{11}, V_{13}, S_9\}$. When the target attack node is set to $S_9$, there are a total of 7 candidate paths from different assumptions ($BX_1$, $B_1$, $B_2$) to $S_9$. Next, three candidate attack path sequences with the highest posterior probability are identified sequentially.

The detailed calculation process is listed in Table 9.

Note that the same path may exist under different attack strategy assumptions. The established attack paths under different assumptions should be corrected according to the maximum posterior probability to describe the reality as accurately as possible. Finally, the attack path sequence with the maximum posterior probability is identified. Three candidate paths can be obtained as follows:

(1) $B_1(BX_1)\rightarrow V_1\rightarrow S_2\rightarrow V_3\rightarrow S_4\rightarrow V_6\rightarrow S_6\rightarrow V_{11}\rightarrow S_8\rightarrow V_{13}\rightarrow S_9$ and its probability is 0.3542;
(2) $BX_1\rightarrow V_2\rightarrow S_3\rightarrow V_9\rightarrow S_9$ and its probability is 0.2941;
(3) $B_1(BX_1)\rightarrow V_1\rightarrow S_2\rightarrow V_5\rightarrow S_8\rightarrow V_{13}\rightarrow S_9$ and its probability is 0.1293.

The three candidate paths with the highest posterior probability calculated are consistent with the candidate paths presented in Ref [25], and the calculating process for other attack scenarios is similar to that of attack scenario 1. The results in Table 7 reveal that DUCAG method

**Table 9** Calculation process of the attack scenario for simulated path 1

| Candidate Paths | Posterior Probability of Attack Paths | Probability Normalization |
|---|---|---|
| $B_1(BX_1)\rightarrow V_1\rightarrow S_2\rightarrow V_5\rightarrow S_8\rightarrow V_{13}\rightarrow S_9$ | 0.0654 | 0.1293 |
| $B_1(BX_1)\rightarrow V_1\rightarrow S_2\rightarrow V_3\rightarrow S_4\rightarrow V_6\rightarrow S_6\rightarrow V_{11}\rightarrow S_8\rightarrow V_{13}\rightarrow S_9$ | 0.1792 | 0.3542 |
| $B_1(BX_1)\rightarrow V_1\rightarrow S_2\rightarrow V_4\rightarrow S_5\rightarrow V_7\rightarrow S_6\rightarrow V_{11}\rightarrow S_8\rightarrow V_{13}\rightarrow S_9$ | 0.0249 | 0.0492 |
| $B_1(BX_1)\rightarrow V_1\rightarrow S_2\rightarrow V_4\rightarrow S_5\rightarrow V_8\rightarrow S_7\rightarrow V_{12}\rightarrow S_8\rightarrow V_{13}\rightarrow S_9$ | 0.0025 | 0.0049 |
| $B_1(BX_1)\rightarrow V_1\rightarrow S_2\rightarrow V_3\rightarrow S_5\rightarrow V_{10}\rightarrow S_8\rightarrow V_{13}\rightarrow S_9$ | 0.0261 | 0.0516 |
| $BX_1\rightarrow V_1\rightarrow S_2\rightarrow V_4\rightarrow S_5\rightarrow V_9\rightarrow S_9$ | 0.0589 | 0.1164 |
| $BX_1\rightarrow V_2\rightarrow S_3\rightarrow V_9\rightarrow S_9$ | 0.1488 | 0.2941 |

can represent the attack scenarios in a manner of being consistent with Ref [25], which validates the rationality of DUCAG and the effectiveness of attack scenario construction.

### Experiment 3: comparison of risk probability calculation methods

To quantify the dynamic risk probability of nodes in the DUCAG, this study refers to the actual network and Bayesian attack graph presented in Ref [35] and updates the Bayesian attack graph model using the modeling method of DUCAG. The proposed CCRP algorithm is compared to the methods presented in Refs [32, 35, 36]. The Bayesian attack graph of the network constructed in Ref [35] is shown in Fig. 7.

By updating the structure of the Bayesian attack graph in the Fig. 7 (a) based on DUCAG, the attack graph is obtained as shown in the Fig. 7 (b), in which different $B$ variables represent different attack strategies adopted by the attacker in the initial attack stage. In experiment 3, the root permission of the workstation is set as the attacking target, and the corresponding state node is $S_7$. The intrusion process of attack target 1 is simulated, and the intrusion detection system detects the attacker's attack evidence $o_1$ and $o_4$. In the attack graph, the corresponding state nodes are $V_1$ and $V_4$. Under the above evidence, the simplified DUCAG obtained is shown in Fig. 8 in which only the $B_2$ assumption can explain all the detected attack evidence.

In this experiment, two alarm data, $o_1$ and $o_4$, are observed. The alarm information is utilized as evidence to calculate the risk probabilities of various nodes in the network by applying the CCRP algorithm. The results of related works listed in Table 10 are according to Ref [35], and in contrast, the outcomes of CCRP algorithm in this paper are also listed. The method in Refs [35, 36] considers the confidence level of the evidence and can effectively determine the risk probability of the nodes. The method in Ref [32] considers the interaction between different pieces of evidence and uses a combination of forward and backward updates to calculate the posterior probability of each state node. However, these methods neglect the corroborating effect of downstream evidence on the nodes during the backward update process. By applying the DUCAG and the corresponding node risk probability update algorithm, the posterior probability of each state node can be calculated, considering the initial attack hypothesis and the relationship between the evidence. Based on Table 10 and Fig. 8, the most possible attack path for the attacker is $B_2 \rightarrow V_1 \rightarrow S_1 \rightarrow V_4 \rightarrow S_4 \rightarrow V_8 \rightarrow S_7$, which is consistent with the results presented in the Refs [34, 35]. Therefore, the proposed method in this study can effectively evaluate the risk of state nodes in a network by considering the relationship between evidence and hypothesis, and identify the attacker's intrusion intent in a more reasonable way.

Based on the experimental comparison method presented in Ref [33], Table 11 lists a comprehensive

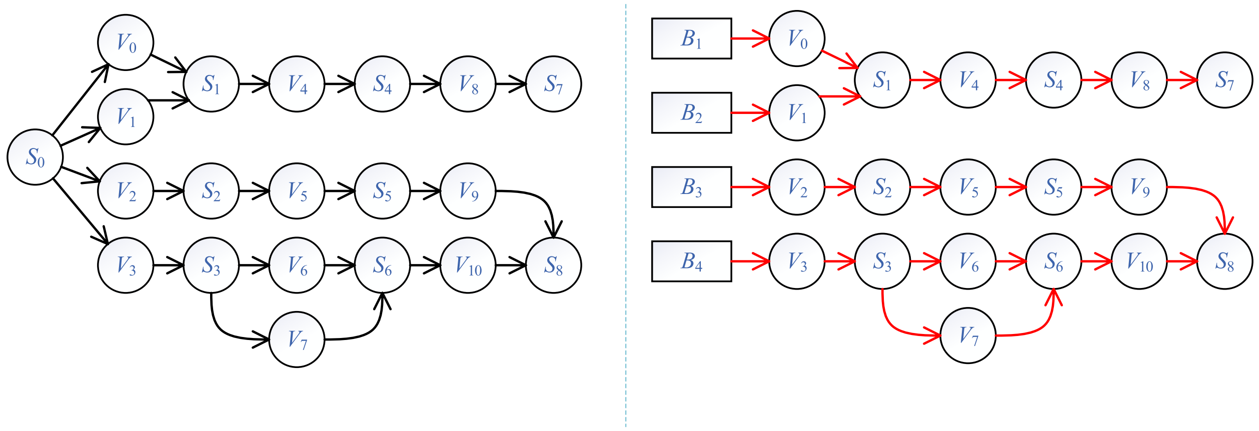

**Fig. 7** **a** The Bayesian attack graph constructed in [35], **b** The corresponding DUCAG

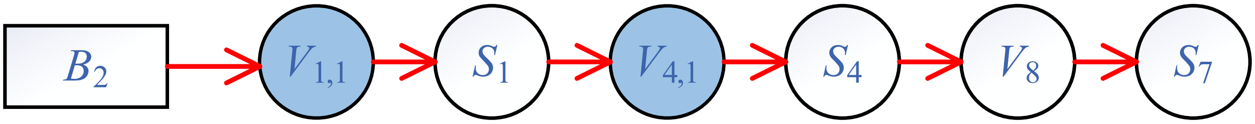

**Fig. 8** The simplified DUCAG of experiment 3 under specific evidence

**Table 10** Comparison table of risk probability by literature node

| Methods of Reasoning | Dynamic Risk Probability for Each State Node | | | |
|---|---|---|---|---|
| | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
| | $S_5$ | $S_6$ | $S_7$ | $S_8$ |
| Node Prior Probability | 0.650 | 0.900 | 0.500 | 0.846 |
| | 0.630 | 0.485 | 0.593 | 0.466 |
| Ref [32] Method | 0.928 | 0.900 | 0.500 | 0.939 |
| | 0.630 | 0.485 | 0.657 | 0.466 |
| Ref [35] Method | 0.970 | 0.900 | 0.500 | 0.979 |
| | 0.630 | 0.485 | 0.685 | 0.466 |
| Ref [36] Method | 1.000 | 0.900 | 0.500 | 0.900 |
| | 0.630 | 0.485 | 0.630 | 0.466 |
| **CCRP Algorithm** | 0.950 | 0.900 | 0.500 | 1.000 |
| | 0.630 | 0.485 | 0.700 | 0.466 |

comparison between the proposed CCRP algorithm in this paper, and other existing risk probability calculation method. Ref [35] focuses on the relevance of evidence, but it is difficult to meet the dynamic requirements of risk probability calculation, and Ref [36] is the opposite. According to Table 11, most of the existing researches mainly focus on calculating the attack paths with the highest probability and the corresponding risk probability values for each path. However, attentions have less been paid to the correlation among evidence. The CCRP algorithm expands all relevant causal chains of a node and then substitutes the prior probability values of the causal chain's related nodes to calculate the posterior probability of that node. In comparison, the CCRP algorithm can dynamically assess the risk probability of network nodes and comprehensively reason about the inherent and uncertain correlations between evidence and hypotheses.

In the application process, the CCRP algorithm can be multi-threaded implemented. In the DUCAG model, the downstream nodes of evidence can store all relevant causal chains of that node in advance, and the posterior probability of the node can be calculated in parallel by applying the CCRP algorithm. From a theoretical perspective, applying the CCRP algorithm to calculate the risk probability of nodes can be achieved by simply traversing the vertex set of the DUCAG. After the DUCAG is simplified, the network scale is greatly reduced, which in turn greatly reduces the amount of computation for attack scenario construction. Therefore, the CCRP algorithm can better meet the real-time requirements in complex cloud computing network environment.

**Table 11** This paper compares the characteristics of probability calculation methods with previous studies

| Features | Shortest Attack Route | Most Possible Attack Route | Evidence Association | Probabilities Updated Dynamically |
|---|---|---|---|---|
| Ref [32] | | | | √ |
| Ref [35] | √ | √ | √ | |
| Ref [36] | √ | √ | | |
| **CCRP Algorithm** | √ | √ | √ | √ |

### Experiment 4: time complexity analysis

Experiment environment: Operating system Windows 11, CPU: 12th Gen Intel (R) Core (TM) i7-12,700. 2.10 GHz, RAM: 16.00 GB (15.7 GB available). Experimental tools: Python 3.10, PyCharm.

The purpose of this experiment is to compare the time complexity of the CCRP algorithm with the Bayesian Network-based risk probability (BNRP) algorithm, which is the fundamental algorithm applied in the above-mentioned researches [24, 35]. In this experiment, the time efficiency of the proposed algorithm is verified by simulating directed acyclic networks of different scales. On the premise that no loops can be formed, each node is randomly added into the graph and establishing directed connecting edges with other existing nodes. The number of evidence nodes in the experiment is set to half of the number of hosts, and 25 groups of experiments are carried out. Figure 9 shows the comparison results of the time complexity of CCRP algorithm and BNRP algorithm under different scales of networks.

Figure 9 (a) shows that the CCRP algorithm has lower time complexity than the Bayesian risk probability inference algorithm under conditions of the same number of evidence and hosts. Figure 9 (b) demonstrates the elapsed time in natural logarithmic scale, and we can see that the CCRP algorithm can show a linear growth trend that can meet the real-time requirements of network security risk analysis. The CCRP algorithm for calculating the risk probability of nodes has the following advantages:

(1) The CCRP algorithm can implement parallel computing. In contrast with Bayesian reasoning, the process of nodes unfolding the causal chain does not depend on each other, so the risk probability of multiple nodes can be calculated at the same time, which significantly improve the computing efficiency.

(2) In CCRP algorithm, the causal paths from one node to others can be pre-calculated and stored as a node set in advance. Thus, once the alarm evidence is overserved in real time in the cloud environment, the CCRP can be performed rapidly by traversing and querying the node set in the attack graph, and the algorithm time complexity of this process is O($N$).

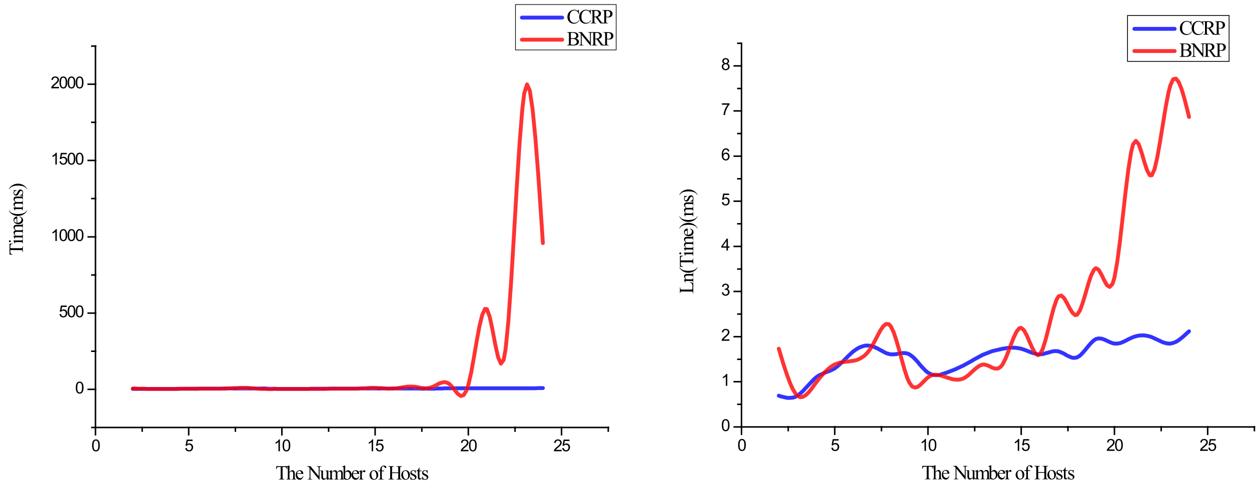

**Fig. 9** **a** Time elapsed results of CCRP and BNRP, **b** Comparisons of CCRP and BNRP in Natural Logarithmic Scales

So the CCRP algorithm can better meet the requirements of high time efficiency for the applications on cloud environment.

## Conclusions

With the development of cloud computing technology, network security issues are becoming increasingly prominent. Attack behaviors in the cloud computing environment exhibit characteristics such as complexity and diversity. It has been a challenge to accurately characterize attack behaviors and achieve efficient and real-time inference. Most of current attack graph modeling methods are based on Bayesian networks. However, attentions have less been paid to the correlation among evidence, and high reasoning complexities reduce the practicality of these methods. Therefore, this paper proposes a novel probabilistic graphical model, DUCAG, to represent the uncertain causalities in complex attack scenarios. Furthermore, a CCRP algorithm is designed to dynamically update the causality weights among different network attack evidence events and attacker hypotheses in real-time, thereby predicting the risk probabilities of each node. Experiments reveal that the proposed method can effectively model the attack scenarios and underlying causalities, and efficiently predict the attacker's capabilities and potential attack likelihood under different attack modes. Benefitting from the proposed causal chain-based logical reasoning mechanism, the CCRP algorithm can better meet the requirements of high time efficiency for the applications on cloud environment and is robust to the incompleteness and redundancy in alarm information. Therefore, the proposed method has application significance on cloud computing platforms by evaluating network security status and predicting the risks and behaviors of attackers.

This paper has not yet addressed the issues of automated modeling for attack scenarios and possible loop causality modeling issues among dynamic attack events. To solve these problems, our future work includes:

(1) Designing algorithms that can automatically construct the DUCAG by analyzing historical log data and domain knowledge, and extending to large-scale real cloud platform networks;
(2) Solving the issues of loop causality modeling problems in complex attack scenarios.

### Authors' contributions

Chunling Dong: Conceptualization, Methodology, Writing- Reviewing. Yu Feng: Writing- Original draft preparation, Writing- Reviewing. Wenqian Shang: Corresponding author.

### Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62176240), Beijing Natural Science Foundation (Grant No. 4222038), and Public Computing Cloud, CUC.

## Declarations

### Ethics approval and consent to participate

Not applicable.

### Competing interests

The authors declare no competing interests.

Received: 15 September 2023 Accepted: 7 December 2023
Published online: 24 January 2024

### References

1. Qi L, Xu X, Wu X, Ni Q, Yuan Y, Zhang X (2023) Digital-Twin-Enabled 6G Mobile Network Video Streaming Using Mobile Crowdsourcing. IEEE J Sel Areas Commun 41(10):3161–3174

2. Miao Y, Bai X, Cao Y, Liu Y, Dai F, Wang F, et al (2023) A Novel Short-Term Traffic Prediction Model Based on SVD and ARIMA With Blockchain in Industrial Internet of Things. IEEE Internet Things J 10(24):21217–26
3. Kong L, Li G, Rafique W, Shen S, He Q, Khosravi MR, et al (2022) Time-Aware Missing Healthcare Data Prediction Based on ARIMA Model. IEEE/ACM Trans Comput Biol Bioinform 2022:1–10
4. Wang F, Wang L, Li G, Wang Y, Lv C, Qi L (2021) Edge-cloud-enabled matrix factorization for diversified APIs recommendation in mashup creation. World Wide Web 2021:1–21
5. Wang F, Zhu H, Srivastava G, Li S, Khosravi MR, Qi L (2021) Robust collaborative filtering recommendation with user-item-trust records. IEEE Trans Comput Soc Syst 9(4):986–996
6. Fan Y, Zhao G, Lei X, Liang W, Li K-C, Choo K-KR et al (2021) SBBS: A secure blockchain-based scheme for IoT data credibility in fog environment. IEEE Internet Things J 8(11):9268–9277
7. Fan Y, Zhang W, Bai J, Lei X, Li K (2023) Privacy-preserving deep learning on big data in cloud. China Communications 20(11):176–186
8. Mahenge MPJ, Li C, Sanga CA (2022) Energy-efficient task offloading strategy in mobile edge computing for resource-intensive mobile applications. Digital Communications and Networks 8(6):1048–1058
9. Qi L, Lin W, Zhang X, Dou W, Xu X, Chen J (2023) A Correlation Graph Based Approach for Personalized and Compat-ible Web APIs Recommendation in Mobile APP Development. IEEE Trans Knowl Data Eng 35(6):5444–57
10. Song W, Wu Y, Cui Y, Liu Q, Shen Y, Qiu Z et al (2022) Public integrity verification for data sharing in cloud with asynchronous revocation. Digit Commun Netw 8(1):33–43
11. Han H, Fei S, Yan Z, Zhou X (2022) A survey on blockchain-based integrity auditing for cloud data. Digit Commun Netw 8(5):591–603
12. Yang Y, Yang X, Heidari M, Khan MA, Srivastava G, Khosravi MR et al (2023) ASTREAM: Data-Stream-Driven Scalable Anomaly Detection With Accuracy Guarantee in IIoT Environment. IEEE Trans Netw Sci Eng 10(5):3007–3016
13. Zhang S, Liu C, Li X, Han Y (2022) Runtime reconfiguration of data services for dealing with out-of-range stream fluctuation in cloud-edge environments. Digit Commun Netw 8(6):1014–1026
14. Xu Z, Zhu D, Chen J, Yu B (2022) Splitting and placement of data-intensive applications with machine learning for power system in cloud computing. Digit Commun Netw 8(4):476–484
15. Kong L, Wang L, Gong W, Yan C, Duan Y, Qi L (2021) LSH-aware multitype health data prediction with privacy preservation in edge environment. World Wide Web 2021:1–16
16. Wang F, Li G, Wang Y, Rafique W, Khosravi MR, Liu G et al (2023) Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city. ACM Trans Internet Technol 23(3):1–19
17. Wang X, Sun Y, Ding D (2023) Adaptive dynamic programming for networked control systems under communication constraints: a survey of trends and techniques. Int J Netw Dyn Intell 2023:85–98
18. Dong C, Zhang Q (2020) The cubic dynamic uncertain causality graph: A methodology for temporal process modeling and diagnostic logic inference. IEEE Trans Neural Netw Learn Syst 31(10):4239–4253
19. Dong C, Zhou J (2023) A new algorithm of cubic dynamic uncertain causality graph for speeding up temporal causality inference in fault diagnosis. IEEE Trans Reliab 72(2):662–677
20. Zhang Q, Dong C, Yan C, Yang Z (2014) Dynamic Uncertain Causality Graph for knowledge representation and probabilistic reasoning: statistics base, matrix, and application. IEEE Trans Neural Netw Learn Syst 25(4):645–663
21. Dong C, Zhou Z, Zhang Q (2018) Cubic dynamic uncertain causality graph: A new methodology for modeling and reasoning about complex faults with negative feedbacks. IEEE Trans Reliab 67(3):920–932
22. Phillips C, Swiler LP (1998) A graph-based system for network-vulnerability analysis. Proceedings of the 1998 workshop on New security paradigms 1998:71–79
23. Xiu-juan W, Bo S, Yan-wen L, Cong-bin X (2015) Computer network vulnerability assessment based on Bayesian attribute network. J Beijing University Posts Telecommunications 38(4):110
24. Poolsappasit N, Dewri R, Ray I (2011) Dynamic security risk management using bayesian attack graphs. IEEE Trans Dependable Secure Comput 9(1):61–74
25. Wang S, Tang G, WANG J (2018) Attack scenario construction method based on causal knowledge net. J Comput Res Develop 55(12):2620–2636
26. Wang W, Du X, SHAN D, (2021) Construction method of attack scenario in cloud environment based on dynamic probabilistic attack graph. J Communications 42(1):1–17
27. Wang L, Islam T, Long T, Singhal A, Jajodia S (2008) An attack graph-based probabilistic security metric. Data and Applications Security XXII: 22nd Annual IFIP WG 113 Working Conference on Data and Applications Security London, UK. Proceedings 22(Springer):283–96
28. Ye Y, Xu X-S, Jia Y, Qi Z-C (2010) An attack graph-based probabilistic computing approach of network security. Jisuanji Xuebao (Chinese J Comput) 33(10):1987–1996
29. Chen F, Zhang Y (2010) Research of quantitative vulnerability assessment based on attack graphs. Comput Eng Sci 32(10):8–11
30. Ammann P, Wijesekera D, Kaushik S (2002) Scalable, graph-based network vulnerability analysis. Proceedings of the 9th ACM Conference on Computer and Communications Security 2002:217–24
31. Liu X (2011) Research on network vulnerability assessment and intrusion alert analysis technology. Ph.D. Thesis, Huazhong Normal University, China
32. Li J, Ling X, Li C, Li Z, Yang J, Zhang L (2022) Dynamic network security analysis based on bayesian attack graph. Computer Sci 49(03):62–69
33. Hu H, Liu Y, Zhang H, Yang Y, Ye R (2018) Route prediction method for network intrusion using absorbing Markov chain. J Comput Res Development 55(4):831–845
34. Jiang N, Cui Y, Wang J, Wu J (2020) Context-based Attack Scenario Reconstruction Model for IDS Alarms. Netinfo Security 20(7):1–10
35. Wang Y, Wu J, Huang J, Hu H, Liu Y (2019) Network Intrusion Intention Recognition Method Based on Bayesian Attack Graph. Comput Eng Appl 55(22):73–79
36. Chen X, Fang B, Tan Q, Zhang H (2014) Inferring attack intent of malicious insider based on probabilistic attack graph model. Chinese J Comput 37(1):62–72

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.