УДК 004.056.55/003.26

МОДЕЛИРОВАНИЕ В РЕШЕНИИ ЗАДАЧ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ С ПОМОЩЬЮ ГРАФОВЫХ БАЗ ДАННЫХ

П.Ю. Филяк (Москва), В.Ю. Латкин (Сыктывкар), Л.А. Виткова (Санкт-Петербург)

Введение и общая постановка задачи

Как следует из определений, приведенных в законодательных и нормативных правовых актах Российской Федерации, национальная безопасность, информационная безопасность Российской Федерации, информационная безопасность организации является состоянием - состоянием защищенности, отсутствием опасности - мгновенным состоянием. Современная система безопасности, как правило, имеет сетевую структуру [1], [2]. Поэтому задачам моделирования сетей применительно к решению проблем безопасности посвящено много исследований, результаты которых отражены в многочисленных статьях и изданиях [1], [3], [4]. Анализ данных, в том числе текстовых, методы и алгоритмы анализа – системный, лингвистический, кластерный и другие могут быть реализованы в разных форматах [1], [2] и с помощью различных инструментов. Для анализа социальных сетей [4], особенно с позиций решения задач информационной безопасности, важнейшим условием является наглядность. Поэтому методы и инструменты визуализации, в частности путем применения графов, становятся особенно актуальными и востребованными в силу следующих причин.

Во-первых, за счет повышения информативности посредством использования графовых моделей – системы узлов и рёбер, в которых задаются и описываются связи и разнообразие отношений. Во-вторых, графовые модели, по определению, являются наглядными и более динамичными. В-третьих, скорость вычислений может быть намного выше, чем при использовании традиционных моделей на основе реляционных баз данных вследствие реализации более рациональных вычислительных процессов.

При решении задач и проблем информационной безопасности в случаях рассмотрения сетевого аспекта отправной точкой и исходным императивом является расчетная схематизация рассматриваемой сети [2] (рис.1).

В зависимости от вида и типа сети исследуется топология с использованием соответствующего математического аппарата, наиболее подходящего для решения задач данного класса.

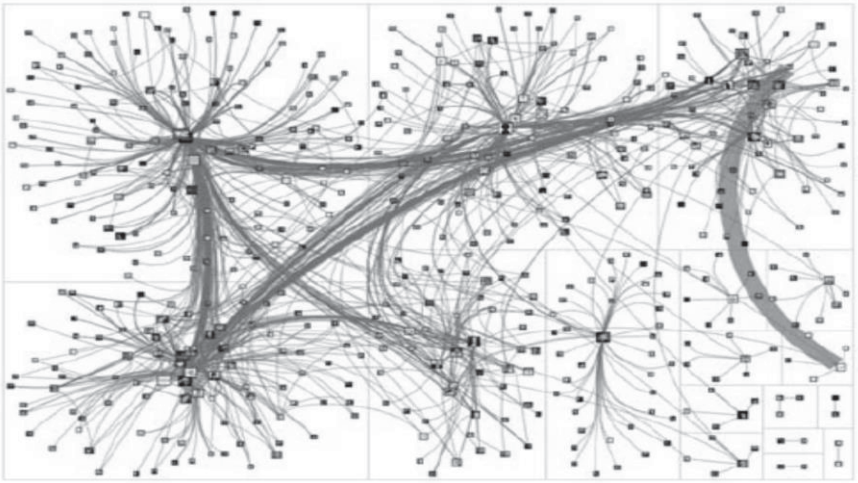

Рис.1. Граф, описывающий сегмент социальной сети

Расчетная схематизация

В первую очередь следует сказать о необходимости применения системного подхода и вначале осуществить декомпозицию сети путем выделения из неё элементарных характерных для неё фрагментов.

Теоретической основой выбранного подхода является использование дискретной математики и, в частности, её раздела – теории графов [5]. Соответственно, вначале производится расчетная схематизация, которая в дальнейшем описывается математически, и далее применяются инструментальные средства, основанные на широком использовании современных информационных технологий, алгоритмических языков, методов, средств и сред программирования (рис.2).

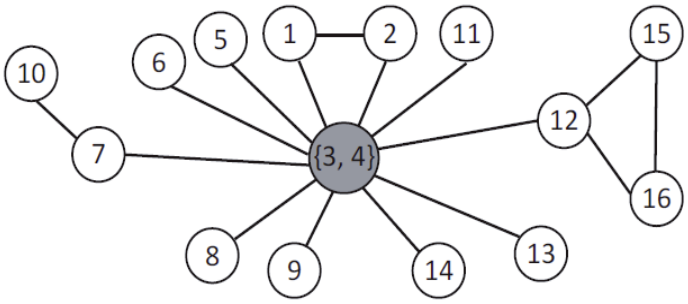

Рис.2. Формирование фрагмента графа сети

Декомпозиция социальной сети

Рассмотрим социальную сеть. Выберем фрагмент с определенным количеством персонажей. Возьмем, для примера, количество персонажей в пределах примерно десяти, в нашем случае будет двенадцать. С одной стороны, 10-12 человек – это некое математическое ожидание, по статистике, средней номинальной группы лиц по интересам, с другой стороны, количество связей, отображаемых на создаваемом графе фрагмента социальной сети будет такоим, что полученный граф будет достаточно наглядным для изображения связей между членами группы, а не превратится на рисунке, воспроизводящем его, в «клубок» с запутанными нитями, в котором трудно будет разглядеть характерные взаимоотношения между участниками сети.

Участники группы будут смоделированы в виде узлов графа, а связи между ними – представлены так называемыми ребрами графа, на которые будут наложены дополнительно и отношения между ними, которые раскрывают практически неограниченный спектр возможных вариантов общения между лицами. Общее количество связей между участниками сети (элементами системы) подсчитывается по тривиальной формуле:

$$C = n * (n-1),$$

где C – количество двусторонних связей между элементами системы (сети) – группы – между участниками (членами) сети;

n – количество участников (членов) группы в сети.

Для примера создадим сеть из двенадцати участников (членов), которые в графовой модели сети будут представлены узлами с соответствующей информацией о каждом из участников в соответствующих узлах. Зададим участникам разные имена и какие-либо дополнительные характерные параметры по каждому из них, чтобы можно было идентифицировать каждый персонаж. В данном конкретном случае в качестве таких данных будем использовать год рождения и город, из которого они приехали (живут). В общем случае, если данные по используемым параметрам (имена, год рождения, город,…) будут совпадать, необходимо ввести какой-нибудь дополнительный

параметр, чтобы персонально идентифицировать каждого участника, например: Александр I, Александр II и т.д.

В нашем примере участники будут иметь следующие идентификаторы: Александр из города Москва, Александр из Москвы, Егор – Москва, Иван – Санкт Петербург, Илона – Москва, Марк – Москва, Николай – Санкт Петербург, Светлана – Москва, Семен – Калуга, Сергей – Москва, Фома – Рязань, Юрий – Кострома.

Как видно, два участника группы с именем Александр проживают в одном и том же городе – Москве, поэтому для них вводим дополнительный параметр в качестве идентификатора – год рождения. У одного из них – 2000 год, у другого 2002. Для более удобного оперирования персонами при систематизации и построении сети назовем их условно Александр I (2000 года рождения) и Александр II (2002 - г.р.) соответственно.

Для того, чтобы смоделировать и построить сеть, в качестве связей мы зададим им тип «знают» (KNOWS). Это связи двустороннего типа, в противном случае, без наличия таких связей, социальная сеть из обозначенных участников будет несостоятельной, то есть связи типа «знают» (KNOWS) – это MinMin (Minimum minimorum – Минимум миниморум) для существования и соответственно создания социальной сети. Построение социальной сети помимо формирования сети, как таковой, путем создания основных связей между элементами (участниками группы) по типу «знают» (KNOWS), по сути каркаса, предполагает, кроме того, еще и задание для членов сети дополнительных отношений между ними. Для иллюстрации хрестоматийности установления возможных отношений в социальной сети представим в качестве примера следующую их типологию, вариации и комбинации.

«Знают» (Knows) ↔ ; «Испытывают симпатии» (Like) → , ← ; «Дружат» (Be Friends) ↔ ; «Любовь» (Love) →, ↔, ← ; «Испытывают неприязнь» (Dislike) →, ↔, ←; «Ненавидят» (Hate) →, ↔, ← ; «Враждуют» (Feud) ↔ ; «Долг» (кому – указывает стрелка) и сумма (Debt/Credit/Duty) →, ← ; «Помогает» (учеба, физическая, дела) (Help/Assist/Succor) →, ↔, ← ; «Криминальные» (Criminal) →, ↔, ← .

Стрелки: →, ↔, ← , в зависимости от типа отношений могут быть направлены от одного участника к другому либо в обе стороны, что свидетельствует о взаимности.

Для того, чтобы сложилось целостное представление о системе взаимоотношений в сети и их вариативности возможные отношения между участниками группы были сведены в таблицу формализации связей и отношений в проектируемой сети (табл. 1). В табл. 1 имена участников приведены сокращенно для более компактного их отображения.

Таблица 1
Формализация связей и отношений в проектируемой сети

Типы связей	Ал-р I 2000	Ал-р II 2002	Егор	Иван	Илона	Марк	Николай	Светлана	Семен	Сергей	Фома	Юрий
Знают (Knows) ↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔
Испытывают симпатии (Like) →←	→Св	→Св	→Св	→Св	→Св	→Св				→Св		→Св
							Св→		Св→		Св→	
	→Ил	→Ил	→Ил	→Ил		→Ил	→Ил		→Ил	→Ил	→Ил	→Ил
	Ил→	Ил→	Ил→	Ил→		Ил→	Ил→	Ил→	Ил→	Ил→	Ил→	Ил→
	→Ег	Ив↔	→Марк	Ал II ↔		→Сем Серг↔			→Юр	Марк↔	→Ег	→Ник
Дружат (Be Friends) ↔		Ив↔		Ал II ↔		Серг↔				Марк↔		
Любовь (Love) →↔←	→Ил	→Св	→Ил	→Св	→Ник		Св→ →Ил Ил↔	→Ник	→Ил		→Ил	
Испытывают неприязнь (Dislike) →↔←		→Ник →Семен →Фома		→Ник →Семен →Фома								
Ненавидят (Hate) →↔←								→Ил				
Враждуют (Feud) ↔		Ник↔		Фома↔								
Долг (кому – указывает стрелка) и сумма (Debt/Credit/Duty) →←		→Ив				→Серг			→Юр			
Помогает (учеба, физическая, дела) (Help/Assist/Succor) →↔←		→Св		→Св					→Юр			
Криминальные (Criminal) →↔←						→Серг						

Методы и инструменты программирования

Спектр располагаемых инструментов, позволяющих моделировать социальные сети в графическом формате, в настоящий момент достаточно широк. Это, в частности, пакет для сетевого анализа Gephi, инструмент имитационного моделирования AnyLogic [2] и другие. В качестве метода программирования, как и в других авторских случаях [1], [4], был избран подход, основанный на создании собственного программного комплекса, путем программирование в графовых базах данных (ГБД) – система управления графовыми базами данных (СУГБД), а инструментом (средством) программирования избрана ГБД Neoj4j (https://neo4j.com), В ней применяется декларативный язык запросов Cypher, а также Data Definition Language (DDL) – язык определения данных, используемый для синтаксических обозначений, описания структуры базы данных (БД), для создания и удаления объектов.

На языке DDL в Cypher с помощью запросов становится возможным:

- создать узел или отношение (с метками и атрибутами или без них);
- создать индексы – скрытые структуры в СУБД, предназначенные для оптимизации (ускорения) поиска данных;
- удалить существующий объект БД;
- обновить атрибуты узла или отношения. В данную группу запросов входят:
- CREATE – определение и создание объекта;
- MERGE – создание нового объекта или его изменение;
- DELETE – удаление существующего объекта.

Для реализации сети и построения графа можно использовать любую из актуальных на сегодняшний день версий: Neo4j 2.3.1 – наиболее ранняя из действующих, и Neo4j 5.16.0 – самая последняя версия разработки, указанной ГБД.

Принципиальной разницы между версиями нет, есть небольшие отличия в интерфейсах пользователя, формате и параметрах некоторых командных кодов для ввода и вывода информации в графическом и табличном виде, форме и виде узлов и ребер графов. Есть некоторое расширение возможностей в более поздней версии программного продукта по отношению к более ранней. Хотя, справедливости ради следует отметить, что для начинающего программировать в ГБД версия Neo4j 2.3.1 на первых порах будет более приемлемой в силу того, что она проще и интерфейс пользователя в ней интуитивно более понятный.

Построение сети

Построение сети сводится к созданию графа с использованием, как было сказано выше, языка запросов Cypher и языка определения данных – Data Definition Language (DDL). До начала создания графа необходимо создать некую сущность, то есть, выражаясь простым языком, создаваемому графу надо прежде всего дать его название (имя).

В нашем случае обозначим эту сущность идентификатором People (Люди), что будет означать сеть, элементами которой – узлами - будут люди, то есть сеть людей, или социальная сеть.

В общем случае узлами (элементами сети) может быть что угодно: приборы, устройства, технические объекты, любые объекты, люди и объекты, люди и технические объекты (то есть социотехнические системы), состояния, факты, события и т.д. Реализации графа сети в командных кодах для удобства сведена в табл. 2 и снабжена графическими иллюстрациями.

Таблица 2
Реализации графа сети в командных кодах Neo4j

№	Команда (командный код)	Пояснение
Работу в ГБД Neo4j целесообразно начинать «с чистого листа», то есть удалить все ранее существовавшие базы данных, все графы и их сущности для операций во фреймах		
1.	MATCH (n) DETACH DELETE n	Удаление всех сущностей, если таковые не удалены из БД
2.	CREATE (n:People {name: "Александр", from: "Москва", born: 2002})	Создается личность Александр (годы рождения разные - 2002)
3.	MATCH (n) RETURN n	Вывод данных на экран узла (узлов)
4.	CREATE (n:People {name: "Александр", from: "Москва", born: 2000})	Создается личность одного из участников сети с именем «Александр» (годы рождения разные - 2000)
6.	CREATE (n:People {name: "Николай", from: "С-Петербург"})	Создается узел - личность Николай
7.	CREATE (n:People {name: "Семён", from: "Калуга"})	Создается личность Семён
8.	CREATE (n:People {name: "Фома", from: "Рязань"})	Создается личность Фома
9.	CREATE (n:People {name: "Юрий", from: "Кострома"})	Создается личность Юрий
10.	CREATE (n:People {name: "Иван", from: "С-Петербург"})	Создается личность Иван
11.	CREATE (n:People {name: "Марк", from: "Москва"})	Создается личность Марк
12.	CREATE (n:People {name: "Егор", from: "Москва"})	Создается личность Егор
13.	CREATE (n:People {name: "Сергей", from: "Москва"})	Создается личность Сергей
14.	CREATE (n:People {name: "Илона", from: "Москва"})	Создается личность Илона
15.	CREATE (n:People {name: "Светлана", from: "Москва"})	Создается личность Светлана
16.	MATCH (n) RETURN n	Вывод данных (узлов – личностей) на экран
17.	MATCH (ivan: People {name:"Иван"}) MATCH (n: People) WHERE n.name<>"Иван" CREATE (ivan)-[:KNOWS]->(n)	Создание отношений: Иван знает всех

Таблица 2 (продолжение)
Реализации графа сети в командных кодах Neo4j

№	Команда (командный код)	Пояснение
18.	MATCH (n) RETURN n	Вывод данных – узлов (личностей) и созданных связей на экран
19.	MATCH (ivan: People {name:"Иван"}) MATCH (n: People) WHERE n.name<>"Иван" CREATE (ivan)<-[:KNOWS]-(n)	Создание связей: все знают Ивана
20.	MATCH (n) RETURN n	Вывод данных – узлов (личностей) и созданных связей на экран

В итоге получаем промежуточный результат – фрагмент полного графа социальной сети (рис.3), отношения между участниками которой заданы априори для примера и представлены в формализованном виде в табл. 1.

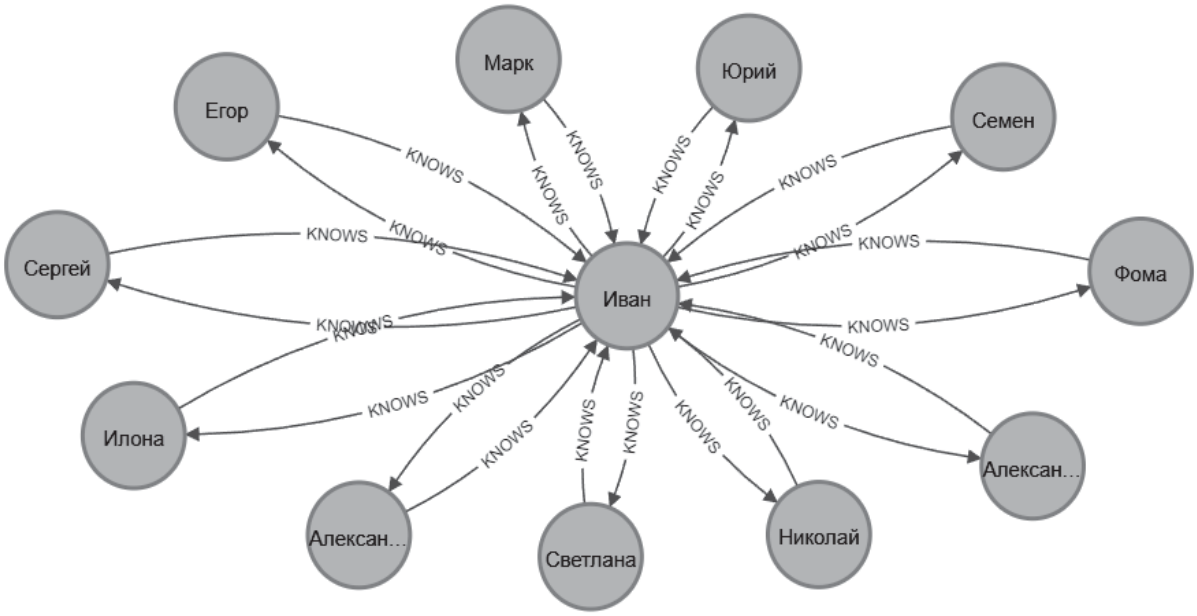

Рис. 3 Фрагмент полного графа социальной сети и связей между участниками

Далее, используя команды языка запросов Cypher и языка определения данных - Data Definition Language (DDL) производится построение полного графа сети, в котором реализованы все связи между её участниками и должны быть заданы все отношения, представленные в табл. 1 (рис.4).

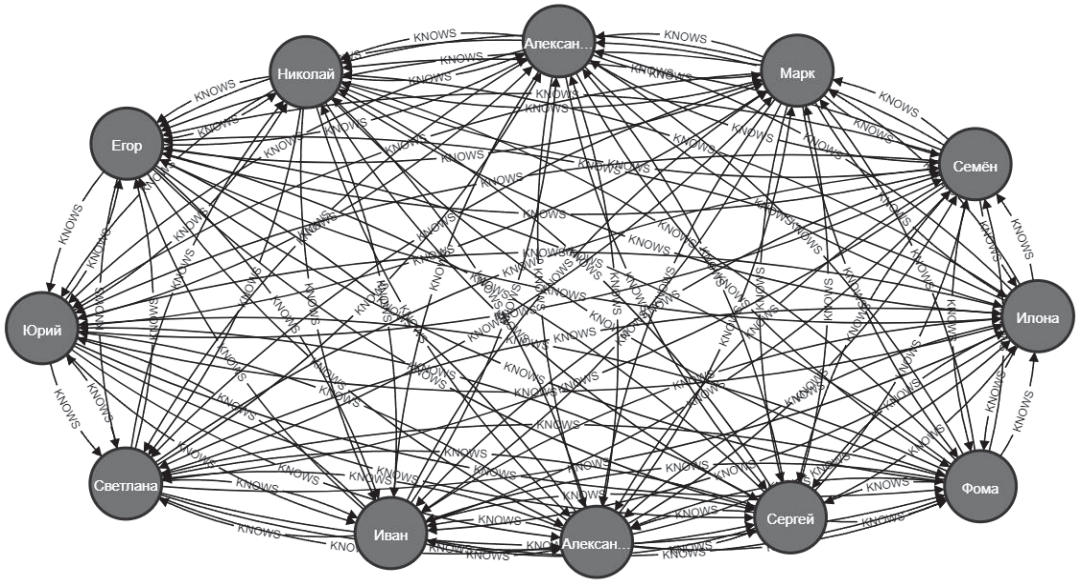

Рис. 4. Полный граф социальной сети и связей между участниками

На рис. 4 представлен итоговый граф, являющийся моделью условной социальной сети, в которой все знают друг друга. На экране монитора во фрейме ГБД Neo4j также дополнительно выводится краткая обзорная информация о созданном графе (рис.5), из которой следует, что сгенерированная сеть состоит из 12 элементов, между которыми в общей сложности существуют 132 взаимные связи.

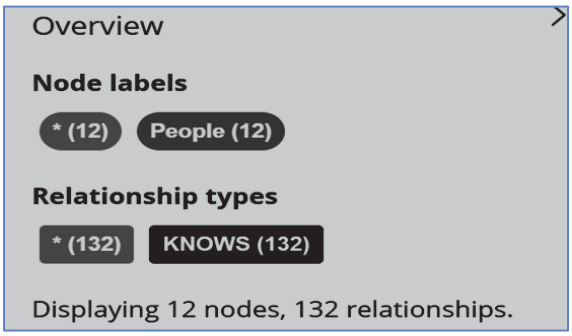

Рис. 5. Обзорная информация о графе

Таким образом, сеть создана, но пока это только «каркас», который необходимо снабдить дополнительной информацией об отношениях участников сети (лиц), что и наполнит графовую базу данными.

Отношения между участниками сети, которые необходимо полностью отобразить в соответствии с табл. 1, также описываются командами языка запросов Cypher и языка определения данных (DDL), в результате чего становятся наглядными и очевидными неформальные связи – симпатий, антипатий, неприязни, отторжения, нетерпимости, любви, зависимостей физических и материальных (долговых обязательств), в том числе и криминальных зависимостей (по тем или иным причинам), которые и предопределяют поведенческую функцию персонажей сети.

Следующей задачей является проверка и анализ данных отношений в сгенерированной социальной сети. Инструментами проверки отношений социальной сети являются команды – запросы системы управления графовой базой данных (СУГБД) Neo 4j. С командами вместе используются: *переменная* - имя переменной узла или отношения; *атрибут* (атрибут узла или отношения); *константа* (представляющие собой числа или строки); *метка.*

В большинстве случаев для верификации сети – узлов и отношений – в качестве команды используется оператор (команда) MATCH с соответствующими параметрами (рис.6).

```
MATCH path = (n:People)-
[r:KNOWS{tyre_friend: "дружат"} ] -
>(m:People) RETURN path
```

Рис. 6. Команда проверки отношений в сети между её участниками

В результате на экран во фрейме будет выведен граф, верифицирующий созданные отношения (рис.7, рис.8)

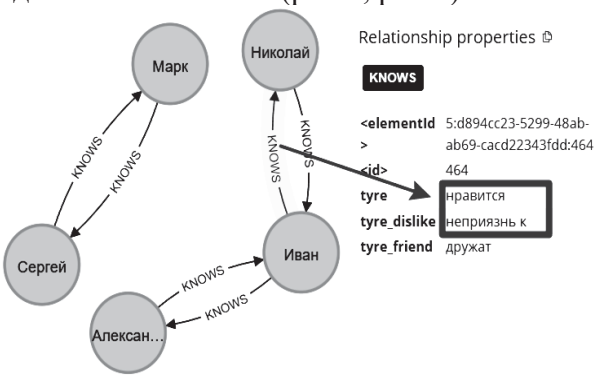

Рис. 7. Граф отношений между персонами

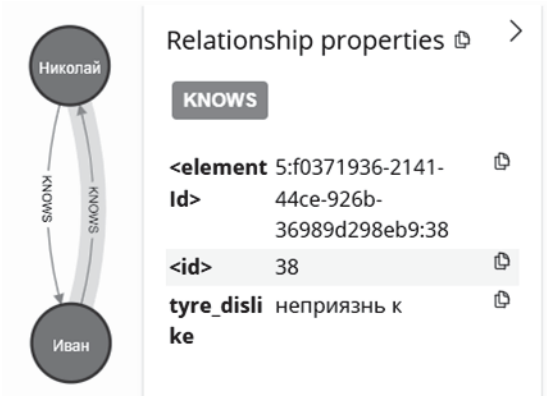

Рис. 8. Граф отношений между персонами во фрейме

Для проверки установленных в сети отношений и информации об участниках сети можно также, помимо команд, эффективно использовать и широкие возможности интерфейса СУГБД Neo 4j, в частности, путем «кликания» клавишей мыши во фрейме на сам узел и его галó – кольцевую ореолу по внешнему контуру круга узла и на ребра сети, соединяющие узлы. При наведении курсора мыши на ребра сети автоматически на экран выводится тип и/или типы отношений между узлами – персонами (участниками) сети (рис.9).

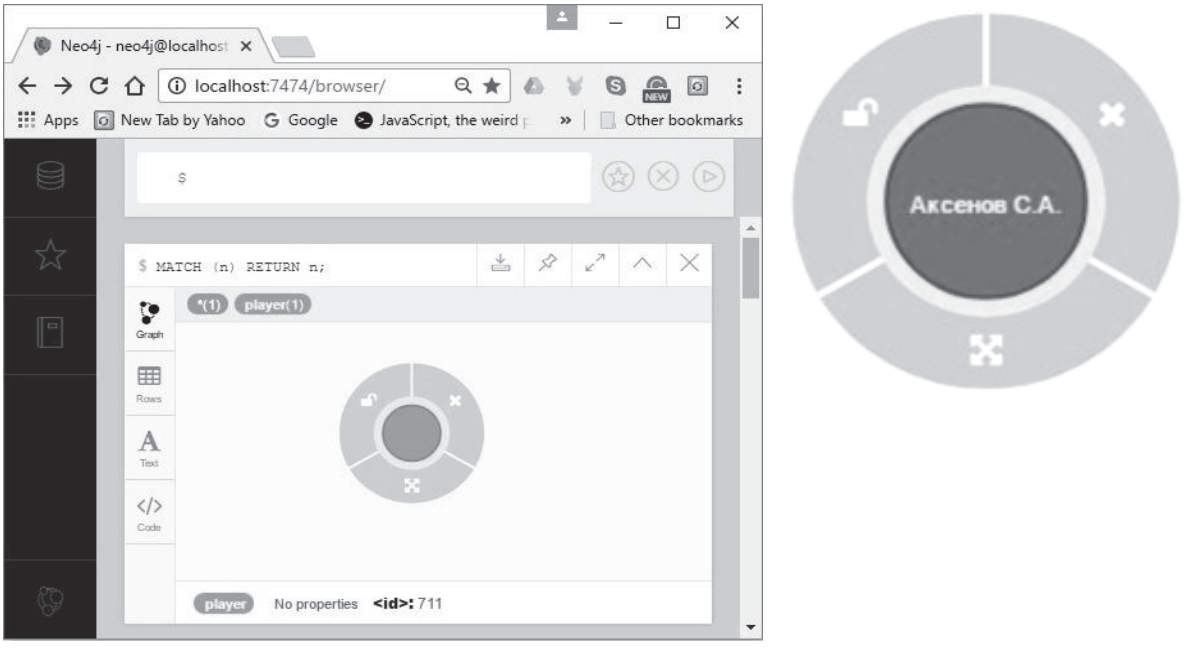

Рис. 9. Узел сети и информационное *галó*

После успешного построения отдельной социальной сети, аналогичным образом создаются и другие социальные сети, а также сети организаций, учреждений, к которым имели, имеют или могут иметь отношение участники отдельных сетей.

Таким образом, путем установления связей между узлами разных сетей – *гетерогенных* сетей - создается единый граф, называемый метаграфом.

Метаграф может адекватно и эффективно моделировать реальные социальные и иные сети, в том числе технические, социотехнические и другие имеющие различную природу, территориальную, пространственную расположенность и ориентацию, а также сети, характеризующиеся таким параметром, как время (рис.10).

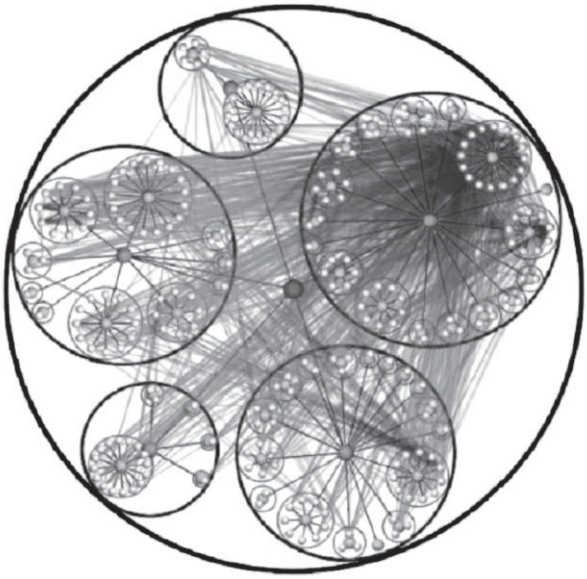

Рис. 10. Сетевой метаграф

Заключение

Как видно из представленных результатов, применение графовых баз данных (на примере Neo 4j) является эффективным инструментом для моделирования сетей различной природы, типа, архитектуры и конфигурации, а также их анализа с точки зрения информационной безопасности и иных позиций. Предлагаемый инструмент позволяет быстро и эффективно с помощью графов построить любую сеть, независимо от её природы и сущности. Создаваемые таким образом модели обладают:

- наглядностью; динамичностью;
- адаптивностью; совместимостью;
- высокой скоростью обработки и анализа данных:
- масштабируемостью; преемственностью;
- легко настраиваются по любому из задаваемых характерных параметров;
- располагают удобным, дружественным и интуитивно понятным интерфейсом.

Технология построения сети, представленная в статье, относится к тематике использования методов программирования и моделирования отношений между людьми в социальных группах. Такие модели на основе системного анализа являются эффективным инструментом выявления и противодействия распространению ложной и вредоносной информации, как в социальных сетях, так и для оценки угроз безопасности организации со стороны персонала, поскольку сотрудники организаций могут находиться в самых разнообразных личных отношениях в рамках организации и за её пределами. Личные отношения между сотрудниками организации и их связи с внешней средой заключают в себе огромный потенциал, который может как повышать уровень безопасности за счет сплоченности коллектива в интересах организации, так и содержать серьезные риски в случае деструктивных намерений локального социума.

Представленные модели и возможности обработки и анализа информации на основе графовых баз данных открывают большие перспективы для управления информационной безопасностью, что подчеркивает актуальность предложенного в статье подхода.

На практике данная задача решается достаточно просто путем использования сервисов социальной сети. Такие сервисы позволяют вести логирование всей переписки, в частности, в самом минимальном для информативности наборе, по таким параметрам, как: отправитель, получатель, время отправления, время прочтения и ответа и так далее. Более того, у системы управления графовой базой (СУГБД) Neo 4j имеются штатные возможности анализа текста и классификации диалогов по тематикам, которые могут прямо или косвенно иметь отношение к деструктивным. В таких случаях будет строиться граф с широким набором меток – метаданных чатов.

К примеру, логирование переписки может сразу же записываться в специальный CSV – файл, на основе которого могут работать скрипты, генерирующие дополняющие метаграфы.

Для демонстрации логики и системности при реализации сети специально была выбрана тривиальная методика построения сети вручную.

Процедура построения сети, представленная в табл. 2 в виде набора команд для наглядной иллюстрации логики и алгоритма, может быть достаточно просто автоматизирована. Штатные средства СУГБД Neo 4j позволяют автономно сгенерировать скрипты и программные коды (табл. 2) и ввести исходные данные из таблиц связей и отношений (см. табл. 1) путем преобразования этих данных из таблиц в CSV–формат для их последующего импорта в Neo 4j. Также для автоматизации генерирования сети могут быть использованы инструменты и библиотеки языков программирования высокого уровня, например, Python, с которым у СУГБД Neo 4j поддерживается полная совместимость, как одним из предусмотренных внешних приложений.

Таким образом, предложенный подход целесообразно рекомендовать для применения на практике при решении ряда задач информационной безопасности, в частности, связанных с расследованием инцидентов ИБ.

Литература

1. **Батура Т.В., Мурзин Ф.А., Проскуряков А.В.** Программный комплекс для анализа данных из социальных сетей // Программные продукты и системы. 2015. № 4 (112). С. 188-197.
2. **Благов А.В., Рыцарев И.А.** Анализ социальных сетей / – Самара: Издательство Самарского университета, 2020. – 104 с.
3. **Борщев А.В.** Как строить простые, красивые и полезные модели сложных систем // Имитационное моделирование. теория и практика, ИММОД 2013. Казань: Фэн АН РТ, 2013. Т.1. С 21-34.
4. **Остапенко А.Г., Паринов А.В., Калашников А.О. и др.** Социальные сети и деструктивный контент; монография / Под ред. чл.-корр. РАН Д.А. Новикова – М.: Горячая линия – Телеком, 2017. – 276 с.: ил. – (Серия «Теория сетевых войн»: Вып. 3).
5. **Суходолов А.П., Лебедев А.В.** Математические методы в правоохранительной деятельности: вопросы противодействия экстремизму в социальных сетях // Всероссийский криминологический журнал. 2018. Т. 12, № 4. С. 468-475.