Таблиця 2 – Вимірювальний стенд 2 (RTO 1012)

Тип генератора	γ_A	$\gamma_Э$	K_s	γ
Волна-4Р	0,12	0,31	1,15	0,85
Дельта-7	0,05	0,25	1,07	0,93

Результати експериментальних досліджень свідчать, що дані, отримані при використанні двох різних вимірювальних стендів, практично збігаються. Відмінність даних не перевищує декількох процентів.

VI Висновки

Основними методами оцінювання коефіцієнта якості шуму генераторів маскуючих завад, застосовуваних у сфері ТЗІ, є методи, засновані на розкладанні щільності ймовірностей миттєвих значень завади у ряд Еджворта, і методи, засновані на визначенні коефіцієнта якості шуму через ентропію завади. Запропоновано і експериментально підтверджено доцільність оцінювати коефіцієнт якості шуму генераторів маскуючих завад через ентропію. Такий метод характеризується більш простим алгоритмом і може бути реалізованим на практиці не тільки за допомогою спеціалізованих приладів, але також за допомогою сучасних осцилографів з вбудованими процесорами, що дозволяють здійснювати необхідні для цих цілей математичні операції.

Список використаної літератури: ***1.*** *Рубичев Н. А. Оценка и измерение искажений радиосигналов. - М.: «Советское радио», 1978. – 168 с.* ***2.*** *Колмогоров А. Н.,. Фомин С. В. Элементы теории функций и функционального анализа. – М.: ФИЗМАТЛИТ, 2004. – 372 с.* ***3.*** *Левин Б. Р. Теоретические основы статистической радиотехники. Книга первая. - М.: «Сов. радио», 1974. - 552 с.* ***4.*** *Тихонов В. И. Статистическая радиотехника. -М.: «Сов. радио», 1966 - 678 с.* ***5.*** *Крамер Гарольд, Математические методы статистики. - М.: «Мир», 1975. – 720 с.* ***6.*** *Горяинов В. Т., Журавлев А. Г., Тихонов В. И. Примеры и задачи по статистической радиотехнике. – М.: «Советское радио», 1970. – 600 с.* ***7.*** *Дмитриев В. И. Прикладная теория информации. – М.: Высшая школа, 1989. – 320 с.* ***8.*** *Куприянов А. И., Сахаров А. В. Теоретические основы радиоэлектронной борьбы. – М.: Вузовская книга, 2007. – 470 с.*

Володимир Луценко
НТУУ «КПІ», ФТІ

УДК 004.43(031):681.3.01(02)

ВИКОРИСТАННЯ БАЗ ДАНИХ ПРИ ПРОЕКТУВАННІ КОМПЛЕКСНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

Анотація: **Розглянуто проблему визначення структури та властивостей такого середовища проектування комплексних систем захисту інформації, котре передбачає здатність до автоматизації процедури проектування за умови використання специфічних баз даних.**
Summary: **The problem of definition of structure and properties of such environment of designing of complex systems of protection of the information which provides possibility of automation of procedure of designing is considered. The special database is used.**
Ключові слова: **Системи захисту інформації, база даних, етапи проектування, автоматизація проектування.**

I Вступ

Складні системи при своєму проектуванні часто передбачають необхідність прийняття рішень при суперечливих або неповних даних, а також за наявності великої кількості неузгоджених параметрів у вигляді початкових даних про об’єкт проектування. Це стосується і технічного захисту інформації (ТЗІ), а особливо комплексних систем захисту інформації (КСЗІ) для інформаційно-телекомунікаційних систем (ІТКС) та об’єктів інформаційної діяльності (ОІД) [1, 2] у тому числі таких, котрі не мають у своєму складі засобів інформаційних комунікацій, або мають, але без виходу за межі контрольованої зони (КЗ).

В [3] показано, що за таких умов на двох етапах проектування КСЗІ, а саме, на етапі обстеження об’єктів при визначенні дестабілізуючих факторів, котрі створюють загрози, та на етапі переходу від загроз до контрдій, рішення приймаються в умовах, які вимагають залучення елементів творчості від проектувальника, тобто носять суб’єктивний характер. Це особливо стосується великих об’єктів. Так, за [4], тільки перелік назв груп можливих порушень з їх змістовим наповненням є розділеним на 6 класів, що вміщують 36 пунктів та 129 підпунктів. Систематизація порушень та їх віднесення до відповідних дестабілізуючих факторів (ДФ)

з метою подальшого визначення можливих каналів несанкціонованого отримання інформації – взагалі творча задача. Достатньо зазначити при цьому, що за шістьма класами можливих каналів витоку інформації, визначеними є загалом 69 їх видів. Стосовно аналогічних переліків загроз та контрдій, наприклад, у Німецькому стандарті [5], котрий фактично представляє собою каталог переліків можливих загроз та можливих контрдій, зазначеними є 5 груп загроз за 397 ознаками та 6 груп контрдій за 608 ознаками. І це тільки перелік, без деталізації і пояснень щодо сенсу кожного пункту з цих переліків. Зазначене вище та інші аргументи дають підстави щодо необхідності автоматизації процедури проектування КСЗІ. Причому специфіка середовища проектування [3] вимагає залучення відносно складних методів, таких як логіко-семантичні методи прийняття рішень, нечіткі множини або системи інтелектуальної підтримки прийняття рішень при неповністю визначених вхідних даних, моделювання образів об'єктів та ін.

У будь якому випадку, мова йде про математичні, логічні або семантичні процедури над великою кількістю даних у вигляді змістовно-образних масивів. Тобто неминучою є необхідність використання специфічних баз даних (БД) для зберігання, поповнення та взаємодії інформаційних масивів опису об'єктів, можливих дестабілізуючих факторів, загроз та контрдій для об'єктів будь-якої складності.

II Постановка завдання

Головним завданням є створення такої системи проектування КСЗІ, котра дозволяє забезпечити автоматизацію процесу проектування, мінімальну залежність результату проектування від виконавця проекту, підвищення об'єктивності результату проектування, можливість централізації виконання проектів КСЗІ в рамках Держави в руках одного відповідального державного виконавця, здатного та вповноваженого здійснювати не тільки виконавчі функції, а й контрольно-супроводжувальні. Таким чином, автоматично можуть вирішуватися питання єдності проектів в умовах ієрархії підпорядкованих об'єктів та підконтрольним стає аудит безпеки будь якого об'єкту в межах Держави.

Метою роботи є порівняння властивостей специфічних БД проектувальника КСЗІ з існуючими типовими БД, визначення можливості використання типових БД у цілях проектування КСЗІ, дослідження діючих класифікаторів БД з метою визначення місця специфічних БД КСЗІ серед загалом діючих класів БД.

III Бази даних як засіб проектування

Визначимося з питанням щодо того, до якого виду БД можуть відноситися БД, що використовує проектувальник КСЗІ об'єктів захисту (ОЗ). Короткий класифікаційний огляд має визначити БД з такими властивостями:

- БД має бути відкритою, тобто може змінювати чи доповнювати свій склад з часом, у міру отримання даних щодо нових, щойно спроектованих об'єктів;
- БД має використовуватися шляхом обміну даними з ліцензованими замовниками процедур проектування, тобто виконавцями проектів КСЗІ за їх запитом;
- засіб прийняття рішень на визначених етапах проектування має коригувати БД за рахунок доповнення БД результатами поточного проекту (процедура динамічного «донавчання»).

Зв'язок між парами БД, такими як БД властивостей об'єкта та БД дестабілізуючих факторів (ДФ), БД ДФ та БД загроз, БД загроз та БД контрдій здійснюється через засіб прийняття рішень, варіанти якого зазначені у вступному розділі. Таким чином загальний вид зв'язків між БД виглядає так, як на рис. 1.

Рисунок 1 – Структура адресування даних від вхідної до вихідної БД

Поняття БД визначає два відомі підходи до організації інформаційних масивів.

1. Файлова організація та організація у вигляді БД. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків. Характерна риса файлового підходу - вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами (групами осіб), відсутня можливість виявити порушення суперечливості збереженої інформації. Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їх опису і визначити таку організацію збереження даних з обліком існуючих зв'язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу БД.

2. БД може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл-програма", вона явним чином декларується в базі. База даних орієнтована на інтегровані запити, а не на одну програму, як у випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління БД (СУБД).

Структурованість даних в БД передбачає не тільки виділення складових частин БД, а і визначення

зв’язків між ними, а також типізацію елементів зв’язку, при котрій з типом елемента або зв’язку співвідноситься певна семантика та дозволені операції. Таким чином, БД включає схему, або як їх називають метадані, що описують логічну структуру БД у формальному вигляді згідно з метамоделлю.

Так, наприклад, згідно з [6], «Постоянные данные в среде базы данных включают в себя схему и базу данных. Схема включает в себя описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных. База данных включает в себя набор постоянных данных, определённых с помощью схемы. Система управления данными использует определения данных в схеме для обеспечения доступа и управления доступом к данным в базе данных».

З перерахованих, тільки перша ознака є суворою. Друга допускає трактування. Тут роль відіграє загальна практика. Так БД не називають файлові архіви, Інтернет портали, електронні таблиці.

Для випадку проектування КСЗІ зручно, якщо БД – це впорядкований набір алогічно (змістовно-семантично) взаємопов'язаних даних, що використовуються спільно, та призначені для задоволення інформаційних потреб користувачів. Головним завданням БД – є гарантоване збереження значних обсягів інформації та надання доступу до неї прикладній програмі. Таким чином, БД складається з двох частин: збереженої інформації та системи управління нею. Існуючі БД зазвичай використовують логічне пов’язування даних, що не є характерним для проектування КСЗІ.

Записи даних організовують як множину фактів. Вважається, що це забезпечує для користувача ефективний доступ до даних.

Структуровані БД використовують структури даних, тобто структурований опис типу фактів за допомогою схеми даних (модель даних). Модель даних описує об'єкти та взаємовідносини між ними. Існує декілька моделей (чи типів) баз даних; основні: плоска, ієрархічна, мережна та реляційна. Приблизно з 2000 року більше половини БД використовують реляційну модель.

До неструктурованих БД відносяться повнотекстові бази даних, які містять неструктуровані тексти статей чи книг у формі, що дозволяє здійснювати швидкий пошук (як наприклад Вікіпедія) і таке визначення для випадку проектування КСЗІ є незручним.

Часто зустрічається характеристика БД на основі певних параметрів або необхідних вимог, наприклад: значна кількість даних; незалежність даних; відкритий доступ до даних; підтримка транзакцій з гарантією відповідних властивостей; гарантована відсутність збоїв та одночасна робота з багатьма користувачами. Одностайності щодо повноти цих характеристик немає. Однак, для проектування КСЗІ має сенс виділити дві характеристики:

1. значна кількість даних, що семантично пов’язані між собою;
2. гарантована відсутність збоїв з одночасною роботою з багатьма користувачами.

Реалізації БД бувають: комерційні; DB2; Informix; Oracle; SQL Server; Дінай. З відкритим кодом: MySQL; Firebird; PostgreSQL.

Список систем керування БД дуже широкий.

Загальновідомо, що більшість БД належить до реляційного типу. Реляційною називається база даних, у якій всі дані, що доступні користувачеві, організовані у вигляді таблиць, зв'язані між собою, а всі операції, що виконуються з даними, зводяться до дій із цими таблицями.

Для опису БД використовується реляційна алгебра. Згідно з визначенням Wikipedia, це «відгалуження логіки першого порядку, множина відношень, замкнених операторами. Оператори застосовуються до відношень, в результаті застосування отримується нове відношення. В математиці, алгебра відношень є алгебраїчною структурою щодо математичної логіки та теорії множин. Подібно до інших алгебр, деякі оператори є примітивними, а інші, будучи визначеними через примітивні, є похідними від них. В реляційній алгебрі Кодда визначено такі шість примітивних операторів: вибірка, проекція, декартів добуток, об'єднання та різниця і перейменування (насправді, Кодд відмовився від включення оператора перейменування, однак, розробники ISBL навели приклади необхідності його включення). Шість операторів є фундаментальними в тому сенсі, що жоден із них не можна відкинути без втрати потужності. Багато інших операторів було визначено комбінацією цих шести. Серед найважливіших можна назвати: перетин множин, ділення та природне об'єднання. Насправді, ISBL дала підстави для заміни декартового добутку природнім об'єднанням, окремим випадком якого є декартів добуток».

Операції з множинами визначаються як декартів добуток, об'єднання множин, різниця множин. Реляційна алгебра користується ще й більш специфічними операторами – узагальнена вибірка як унарний оператор, проекція, перейменування.

Реляційна модель даних – це логічна модель даних. Вперше була запропонована британським ученим з компанії IBM Едгаром Франком Коддом (E. F. Codd) в 1970 році в статті «A Relational Model of Data for Large Shared Data Banks» (така модель вперше була описана в російському перекладі статті в журналі «СУБД» N 1 за 1995 р.). В даний час ця модель є фактичним стандартом, на який орієнтуються практично всі сучасні комерційні СУБД.

У реляційній моделі досягається набагато більш високий рівень абстракції даних, ніж в ієрархічній або мережевій. У згаданій статті Е. Ф. Кодда зазначено, що «реляційна модель надає засоби опису даних на основі тільки їх природної структури, тобто без потреби введення будь-якої додаткової структури для цілей машинного представлення». Іншими словами, подання даних не залежить від способу їх фізичної організації. Це забезпечується за рахунок використання математичної теорії відносин (сама назва «реляційна» походить від англійського relation— «відношення»), що дуже вигідно для проектування КСЗІ.

До складу реляційної моделі даних зазвичай включають теорію нормалізації. Крістофер Дейт визначив три складові частини реляційної моделі даних: структурна, маніпуляційна, цілісна.

Структурна частина моделі визначає, що єдиною структурою даних є нормалізоване *n*-арне ставлення. Відносини зручно представляти у формі таблиць, де кожен рядок є кортеж, а кожен стовпець є атрибут, визначений на деякому домені. Даний неформальний підхід до поняття відносини дає більш звичну для розробників і користувачів форму представлення, де реляційна база даних являє собою кінцевий набір таблиць.

Маніпуляційна частина моделі визначає два фундаментальних механізми маніпулювання даними. Це реляційна алгебра і реляційне числення. Основною функцією маніпуляційної частини реляційної моделі є забезпечення заходів реляційності будь-якої конкретної мови реляційних БД: мова називається реляційною, якщо вона має не меншу виразність і потужність, ніж реляційна алгебра або реляційне числення.

Цілісна частина моделі визначає вимоги цілісності сутностей і цілісності посилань. Перша вимога полягає в тому, що будь-який кортеж будь-якого відношення відмінний від будь-якого іншого кортежу цього відношення, тобто іншими словами, будь-яке відношення має володіти первинним ключем. Вимога цілісності щодо посилань, або вимога зовнішнього ключа полягає в тому, що для кожного значення зовнішнього ключа, що з'являється в відношенні, що посилається, а також у відношенні, на яке ведеться посилання, повинен знайтися кортеж з таким же значенням первинного ключа, або значення зовнішнього ключа повинно бути невизначеним (тобто ні на що не вказувати). Таким чином, для проектування КСЗІ ключем має стати базис структур об’єктів захисту з відношеннями у вигляді логіки зв’язків між складовими базису.

Можна провести аналогію між елементами реляційної моделі даних і елементами моделі «сутність-зв'язок». Реляційні відносини відповідають наборам сутностей, а кортежі — сутностям. Тому, як і в моделі «сутність-зв'язок», стовпці в таблиці, що представляє реляційне відношення, називають атрибутами.

Кожен атрибут визначений на домені, тому домен можна розглядати як безліч допустимих значень даного атрибуту. Кілька атрибутів одних відносин і навіть атрибути різних відносин можуть бути визначені на одному і тому ж домені.

Іменована безліч пар «ім'я атрибута — ім'я домену» називається схемою відношення. Потужність цієї множини називають ступенем чи «арністю» відносини. Набір іменованих схем відносин представляє схему бази даних.

Атрибут, значення якого однозначно ідентифікує кортежі, називається ключовим (або просто ключем). У нашому випадку ключем є атрибут – «номер структурної одиниці базису об'єкта захисту (ОЗ)», оскільки його значення унікально для кожного користувача ключів на ОЗ. Якщо кортежі ідентифікуються тільки зчепленням значень декількох атрибутів, то говорять, що відношення має складовий ключ. Ставлення може містити кілька ключів. Завжди один із ключів оголошується первинним, його значення не може оновлюватися. Всі інші ключі відносини називаються можливими ключами.

На відміну від ієрархічної і мережної моделей даних, в реляційній відсутнє поняття групових відносин. Для відображення асоціацій між кортежами різних відносин використовується дублювання їх ключів.

Переваги реляційної моделі – простота і доступність для розуміння користувачем. Єдиною використовуваною інформаційною конструкцією є: «таблиця»; строгі правила проектування, які базуються на математичному апараті; повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні; для організації запитів і написання прикладного програмного забезпечення немає необхідності знати конкретну організацію БД у зовнішній пам'яті.

Недолік реляційної моделі у тому, що далеко не завжди предметна область може бути представлена у вигляді «таблиць», а в результаті логічного проектування з'являється безліч «таблиць», що є характерним для проектування КСЗІ ОЗ. Це призводить до труднощів розуміння структури даних. БД займає відносно багато зовнішньої пам'яті і низька швидкість доступу до даних.

Зазначений недолік компенсується проектувальником КСЗІ ОЗ шляхом заміни семантичних методів прийняття рішень засобом прийняття рішень. Це дає змогу моделі проектування складати необхідні таблиці даних при кожному акті проектування. Для алгоритмованих моделей така процедура вимагала б нескінченної потужності комп'ютера.

Нормалізація схеми бази даних – це покроковий процес розбиття одного відношення (на практиці таким відношенням є таблиця) відповідно до алгоритму нормалізації на декілька відношень на основі функціональних залежностей.

Нормальна форма — властивість відношення в реляційній моделі даних, що характеризує його з точки зору надмірності, яка потенційно може призвести до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовольняти відношення. Таким чином, схема реляційної бази даних переходить у першу, другу, третю і так далі нормальні форми. Якщо відношення відповідає критеріям нормальної форми рівня *n*, та всіх попередніх нормальних форм, тоді вважається, що це відношення знаходиться у нормальній формі рівня *n*.

Нормальні форми:

Перша нормальна форма (1НФ, 1NF) утворює ґрунт для структурованої схеми баз даних. Ґрунтом є те, що кожна таблиця повинна мати основний ключ з мінімальним набором колонок, які ідентифікують запис. Уникнення повторень груп (категорії даних, що можуть зустрічатись різну кількість разів в різних записах) правильно визначаючи не-ключові атрибути. Атомарність: кожен атрибут повинен мати лише одне значення, а не множину значень.

Друга нормальна форма (2НФ, 2NF) вимагає, аби дані, що зберігаються в таблицях із композитним ключем, не залежали лише від частини ключа. Схема бази даних повинна відповідати вимогам першої нормальної форми. Дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці.

Третя нормальна форма (3НФ, 3NF) вимагає, аби дані в таблиці залежали винятково від основного ключа. Схема бази даних повинна відповідати всім вимогам другої нормальної форми. Будь-яке поле, що залежить від основного ключа та від будь-якого іншого поля, має виноситись в окрему таблицю.

Нормальна форма Бойса-Кодда (НФБК). Відношення знаходиться в НФБК тоді і лише тоді, коли детермінант кожної функціональної залежності є потенційним ключем. Якщо це правило не виконується, тоді, щоб привести вказане відношення до НФБК, його слід розділити на два відношення шляхом двох операцій проекції на кожну функціональну залежність, детермінант якої не є потенційним ключем. Визначення НФБК не потребує жодних умов попередніх нормальних форм. Якщо проводити нормалізацію послідовно, то в переважній більшості випадків при досягненні 3НФ автоматично будуть задовольнятися вимоги НФБК. 3НФ не збігається з НФБК лише тоді, коли одночасно виконуються такі 3 умови: відношення має 2 або більше потенційних ключів; ці потенційні ключі складені (містять більш ніж один атрибут); ключі перекриваються, тобто мають щонайменше один спільний атрибут.

Четверта нормальна форма (4НФ, 4NF) вимагає, аби в схемі баз даних не було нетривіальних багатозначних залежностей множин атрибутів від будь-чого, окрім надмножини ключа-кандидата. Вважається, що таблиця знаходиться у 4НФ тоді і тільки тоді коли вона знаходиться в НФБК та багатозначні

залежності є функціональними залежностями. Четверта нормальна форма усуває небажані структури даних — багатозначні залежності.

П'ята нормальна форма (5НФ, 5NF, PJ/NF) вимагає, аби не було нетривіальних залежностей об'єднання, котрі б не витікали із обмежень ключів. Вважається, що таблиця в п'ятій нормальній формі тоді і тільки тоді, коли вона знаходиться в 4НФ, та кожна залежність об'єднання зумовлена її ключами-кандидатами.

Нормальна форма домен/ключ. Ця нормальна форма вимагає, аби в схемі не було інших обмежень окрім ключів та доменів.

Шоста нормальна форма. Таблиця знаходиться у 6NF, якщо вона знаходиться у 5NF та задовольняє вимозі відсутності нетривіальних залежностей. Зазвичай 6NF ототожнюють з DKNF.

Таким чином, БД проектів КСЗІ ОЗ є структурованою, зі спеціальною реалізацією, з закритою спеціалізованою системою керування, реляційна, з ненормалізованою схемою БД, близькою до DKNF.

Створення системи автоматизованого проектування КСЗІ є можливим при використанні пам’яті з вибіркою за змістом запиту (інакше називають асоціативною пам’яттю - АП), наприклад, на базі моделі нейроподібної ансамблевої сітки з навчанням.

Після цього завданням роботи стає знаходження доказової бази щодо незворотності такого підходу з одного боку, а з другого, щодо дійсних можливостей реального створення такої системи проектування. Методологія створення такої системи та її використання має забезпечувати головні властивості КСЗІ, а саме – дієвість реалізації та об’єктивність у прийнятті рішень при функціонуванні незалежно від складності об’єкту та умов його життєдіяльності, як і зазначалося вище. Об’єктивність та дієвість у даному випадку передбачають незалежність результату проектування від проектанта та врахування онтології діючих стійких до порушень захисту інформації об’єктів.

Завданням засобів сітьового моделювання з навчанням є виконання функцій АП та подальше прийняття квазіоптимальних рішень. Щодо принципової необхідності розробки нових підходів до проектування КСЗІ сумнівів немає, але відкритим є питання тактики використання автоматизованого проектування. Мається на увазі, що необхідно вирішити питання, – чи необхідно використовувати АП як загальну систему проектування КСЗІ від початкового етапу і до отримання проекту захисту, або більш гнучким є підхід, при якому АП використовується тільки на окремих етапах створення проекту КСЗІ, а ті етапи, що піддаються жорсткій алгоритмізації, можуть обслуговуватися традиційними методами. Наразі питання є відкритим. Але напрацювання з використанням сітьового моделювання на окремих етапах створює передумову до прогресивного розвитку нових напрямків при проектування КСЗІ на інших його етапах.

Оскільки початковий етап проектування КСЗІ вимагає залучення методу експертної оцінки, для об’єктивності висновків експертизи логічним є заміна (повна або часткова) суб’єктів-експертів на технічні засоби прийняття рішень. Найбільш привабливим здається підхід, коли статистичні дані щодо результатів аналізу вразливостей діючих об’єктів складають базу даних (БД стану об’єктів). Шляхом аналізу структури об’єктів, їх складу, призначення, умов існування (далі параметрів) формується бібліотека вхідних для БД (вхідна БД) образів об’єктів (рис. 1). Таким чином, визначений набір параметрів відповідає деякому образу об’єкту. Параметри заповнюють вхідну БД у вигляді конечного набору з $i1$ (див. рис. 1) текстів (слів, фраз, словосполучень). Такий конечний набір має регламентуватися документом ТЗІ. Складність у тому, щоб створити такий набір (фактично, набір параметрів), що задовольняє опису образу будь-якого об’єкту. Принципових проблем при цьому не має виникнути, якщо створювати БД кінцевого але достатнього розміру.

Далі, для кожного вхідного образу об’єкту (вхідної БД) визначається образ захищеності у вихідній БД, аналогічно складений (також у текстовій формі) у вигляді списку вразливостей розміром $i2$. Цей текстовий набір саме формує вихідну БД. Так формується структура послідовності дій, де присутні рецептори як вхідні збудники (параметри образу об’єкту), та ефектори (як вихідний перелік вразливостей) (рис. 1).

Після заповнення обох баз даних при складанні опису фрагментів нового, «незнайомого» об’єкту та пред’явленні цих фрагментів до вхідної БД, з неї вилучається вже занесений відповідний (найбільш близький за змістом) перелік фрагментів – «образ об’єкту».

Цей образ є вибіркою адресування до вихідної БД, за якою з вихідної БД вилучається образ захищеності у вигляді переліку вразливостей. На цьому етапі завдання полягає в тому, щоб реалізувати адресування від вхідної БД до вихідної за асоціативним принципом, тобто за змістом запиту, а не за числовою адресою осередку пам’яті. Таким чином можна реалізувати методику роботи АП. Тут може допомогти і умова конечності списку параметрів, а в результаті, і образів об’єкту. При цьому, перелік вразливостей також має бути конечним, хоча його розмір $i2$ може бути довільним і складатися з комбінацій образів об’єкту. Заповнення обох БД є аналогом процесу навчання. Формування переліку вразливостей при «пред’явленні

незнайомого» об'єкту є аналогом процедури розпізнавання (або впізнавання, коли мова йде про здатність людини).

Використання такої системи дозволяє майже вилучити людський фактор з процедури визначення вразливостей об'єкту. Експерт-людина залишається виключно на етапі збору даних про об'єкт і призначений для складання (бажано якнайбільш об'єктивних) даних, що описують його за такими фрагментами: властивості інформації що циркулює на об'єкті, опис поверхового плану, структуру системи технічного забезпечення, опис прилеглої території, тощо.

Рис. 1 ілюструє не структуру АП, а лише сенс алгоритму процедури користування БД як АП.

Заповнення БД, котра будується на базі АП, часто називають навчанням. Тоді говорять про *АП з навчанням*. Існує декілька відомих алгоритмів навчання. Але, інтуїтивно є природним намагання вилучити обмеження щодо кінцевості списку параметрів та вразливостей. Якщо формувати саме такого проектувальника, то постійне накопичення в БД все більшої кількості нових даних від нових спроектованих об'єктів починає з часом впливати на результати подальшого проектування все з більшою вагою.

Тобто поступово реалізується тактика «забування», згідно з якою накопичення усе більшої кількості нових пред'явлень об'єктів для навчання трансформує БД та взагалі АП так, що старі образи, тобто такі що зустрічаються усе рідше, поступово зникають. Процедурно це виглядає як повільне зменшення ваги зв'язків між вхідною та вихідною БД для окремих комбінацій параметрів об'єкту (образів об'єкту). Система проектування набуває властивості поступової автоматичної адаптації до нових видів об'єктів, нових умов їх існування, нових видів вразливостей. Тобто, в процесі життєдіяльності такий автоматизований проектант залишається відкритим до подальшого «донавчання», тобто розвитку. Він «пам'ятає» історію великої кількості діючих об'єктів і доповнюється властивостями нових об'єктів, котрі він проектує. Саме ці властивості і забезпечують властивості системи автоматичного проектування КСЗІ у вигляді адаптації до умов існування (зміна загальної направленості у методології ЗІ, методичної документації, законодавства, тощо), незалежність від експертів на етапі життєвого циклу, об'єктивну дієвість проектування за рахунок користування БД діючих об'єктів.

IV Висновок

Визначено, що БД проектів КСЗІ ОЗ є структурованою, з спеціальною реалізацією, з закритою спеціалізованою системою керування, реляційна – з ненормалізованою схемою БД, близькою до DKNF. Це означає, що БД КСЗІ є дійсно специфічною і не вкладається повною мірою у будь-яку БД відомого типу за існуючими класифікаторами.

Автоматизація проектування КСЗІ є неможливою без створення такої БД або адаптації деякої існуючої БД до стану, котрий здатний забезпечити зазначені вище властивості. Саме таке завдання наразі вирішується спеціалістами КПІ на кафедрі «Фізико-технічних засобів захисту інформації» факультету «Інформаційної безпеки».

*Список використаної літератури**: 1.** ДСТУ ISO/IEC TR 13335:2003. Інформаційні технології // Настанови з керування безпекою інформаційних технологій. Частина 1: Концепції та моделі безпеки інформаційних технологій. Частина 2: Керування та планування безпеки інформаційних технологій. Частина 3: Методи керування безпекою інформаційних технологій. **2.** ДСТУ 3396.1-96. Захист інформації // Технічний захист інформації. Порядок проведення робіт. **3.** Луценко В. М. Визначення уразливості об'єктів інформаційної діяльності як складова порядку розробки систем захисту інформації / Луценко В. М., Худяков В. О. // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. .-К.: ЧП «ЕКМО», НИЦ «ТЕЗИС» НТУУ «КПИ», 2011. Вип. 2 (21) с. 49–55. **4.** Герасименко В. А. Защита информации в автоматизированных системах обработки даннях / Кн. 1 – М.: Энергоатомиздат, 1994 – 400 с. **5.** IT Baseline Protection Manual. Електронний ресурс: http://www.bsi.bund.de/gshb/English/t/t1000.htm на термін 04.2010 р. **6.** ГОСТ Р ИСО МЭК ТО 10032-2007.*

Віталій Носов, Олександр Манжай
Харківський національний університет внутрішніх справ

УДК 65.012.8+34

ОКРЕМІ АСПЕКТИ ПРОТИДІЇ ІНФОРМАЦІЙНІЙ ВІЙНІ В УКРАЇНІ

Анотація: **Проаналізовано зміст та структуру інформаційного протиборства, наведено визначення ключових понять у даній сфері, визначено характерні риси та мету інформаційної війни. На підставі**