УДК 004.056

Д.О. Семченко[1], О.А. Замула[2]

[1] *Харківський національний університет радіоелектроніки, Харків*
[2] *Харківський національний університет імені В.Н. Каразіна, Харків*

РОЗРОБКА ТА ЗАСТОСУВАННЯ МОДЕЛІ ЗАХИЩЕНОГО КАНАЛУ ПЕРЕДАЧІ ДАНИХ НА ОСНОВІ УПРАВЛІННЯ НЕФУНКЦІОНАЛЬНИМИ ВЛАСТИВОСТЯМИ МЕРЕЖЕВИХ ПАКЕТІВ

Представлено модель захищеного каналу зв'язку, що будується на управлінні нефункціональними властивостями мережевих пакетів. Наведені граничні вимоги, щодо використання розробленої моделі, наведена структура моделі та кожного її компонента. Вказані особливості розробленої моделі та сформульовані пропозиції щодо використання моделі в сучасних телекомунікаційних системах.

Ключові слова: *модель, повідомлення, канал зв'язку, часові затримки, пакет даних, бітчасове значення.*

Вступ

Постановка задачі. Важливим фактором що впливає на розвиток інформаційно-телекомунікаційних систем - є підтримка різноманітних зв'язків між абонентами в мережі з одночасним забезпеченням безпеки цих комунікацій [1]. Дуже часто постає задача передачі коротких повідомлень, які використовуються абонентами та являють собою комерційну таємницю. Саме тому, під час побудови корпоративної мережі, постає питання створення надійного захисту від проникнення порушників у мережу та захисту передачі даних всередині мережі. Такий захист реалізується, в тому числі, на базі протоколів TCP/IP, UDP і стандартних Internet-додатків (e-mail, Web, FTP) [2].

Для вирішення задач надійного захисту ресурсів в ІТС необхідно, щоб методи, моделі, засоби та заходи захисту забезпечували, по-перше, захист інформації під час передачі даних через мережу від відомих атак на основі використання криптографічних алгоритмів перетворення інформації, і, по-друге - скритність самого факту передачі цих даних [4].

Використовуючи абстрактну мережеву модель OSI при розробці моделі захищеного каналу зв'язку, процес аналізу пакетів здійснюється на прикладному рівні, у той час, як вбудова затримок здійснюється на транспортному рівні. Інкапсуляція властивостей між рівнями моделі OSI дозволяє вирішити поставлені задачі та використовувати нефункціональні властивості пакетів для передачі інформації у мережі [2, 3].

Визначення граничних вимог щодо моделі захищеного каналу зв'язку. При розробці моделі захищеного каналу зв'язку, визначимо область застосування моделі та, згідно з цим, сформуємо граничні вимоги, що необхідні для функціонування цієї моделі. Для цього був проведений аналіз вузлів типового каналу зв'язку. Аналіз та отримані коефіцієнти бітових помилок для різних часових вікон і мережевих вузлів з використанням метрики Левенштейна [5], дозволили зробити припущення, що коефіцієнт бітових помилок, для кожного вузла, менше ніж 10% - є прийнятним значенням. Прикладом цього став обраний вузол в Сінгапурі, з круговою затримкою (RTT) в 236 мс. Для розміру часового вікна між пакетами даних в 20 мс, коефіцієнт помилок каналу становить близько 4,5%, що є цілком придатне для багатьох додатків з низькою пропускною здатністю.

Однак гарантовано передбачити затримку не можна у зв'язку з різними факторами, що включають перерахунок маршрутів в результаті адміністративних або аварійних змін в мережі, а також, у зв'язку з можливими навантаженнями на роутери, що призводять до переповнення їх внутрішніх черг. Однак, такі події в мережі виникають відносно рідко, а черги в роутерах цілком обробляються за час, не більше сотен наносекунд. Тому, якщо в мережі між двома абонентами знаходиться близько 15 роутерів (а це можуть бути протилежні сторони планети), то згідно з проведеними розрахунками, для 95% пакетів варіювання затримки буде не більше 5 мс. А для усунення решти 5% втрат буде використана повторна передача одних і тих же даних, що розбиваються на пакети.

Після проведеного аналізу була визначена область застосування, а саме - це канал з низькою пропускною здатністю, що використовуються для вирішення задачі передачі коротких повідомлень.

Граничними вимогами для можливості реалізації розробленої моделі є: наявність пакетів в мережі; висока частота відправки пакетів та час між відправленнями двох пакетів носіїв інформації повинні варіювати в інтервалі як мінімум 20 мс.

Основна частина

Розробка моделі починається з перевірки каналу на придатність передачі повідомлення. На рис. 1 надана схема перевірки каналу.

Рис. 1. Схема перевірки каналу

На рис. 1 наведено параметри:

n – кількість вимірювань тривалостей передачі пакетів;

t_{max} – максимальна тривалість передачі пакетів між абонентами у мережі

t_{min} - мінімальна тривалість передачі пакетів між абонентами у мережі,

Придатність каналу зв'язку визначається нерівністю $(t_{max} - t_{min}) * 8 \le \omega$.

Після перевірки каналу зв'язку на придатність передачі даних, необхідно визначити структуру пердачі інформації цим каналом.

Структура передачі інформації наведена на рис. 2. Якщо символ не передається, то в мережу відправляється тільки FSS (послідовність синхронізації кадрів), CRC (алгоритм знаходження контрольної суми, призначений для перевірки цілісності даних) та кількість відправлених та отриманих пакетів. Це потрібно для того, щоб отримувач зміг встановити факт доставки інформації до відправника.

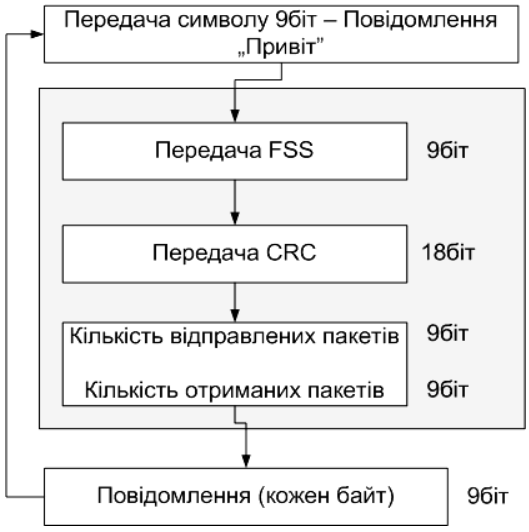

Рис. 2. Структурна схема передачі інформації

Для забезпечення синхронізації даних будемо використовувати дев'яти бітову послідовність, іменовану як FSS. Основною метою FSS є не розбивання шифрованого потоку бітів на кадри, а поліпшення визначення початку послідовності несучих пактів і відкидання усіх попередніх пакетів, які надіслані "клієнтом". Як тільки FSS виявлена в потоці затримок, "клієнт" отримувач стає готовим для декодування корисних даних з затримок. Щоб компенсувати можливі втрати даних закодованих у вигляді затримок, одні й ті ж дані відтворюються повторно в затримках, як проста міра протидії пошкодженню даних, що виникли в результаті мережевого шуму. Як тільки всі необхідні закодовані дані відправлені через прихований тимчасової канал, передачу даних слід повторити, починаючи з FSS. Це повинен враховувати "клієнт" отримувача.

FSS повинна обиратися таким чином, щоб запобігти її появі у безперервному потоці біт. Якщо на цей потік біт не накладати ніяких обмежень, то він потенційно може містити будь-яку послідовність. Таким чином, для запобіганні появи FSS в цьому потоці, застосовується модифікація потоку, що називається Bit Stuffing. На Рис 3 показано, як в результаті Bit Stuffing встановлюється нульовий біт в середину октету. Якщо при цьому вибрати FSS, що складається з 9 одиничних біт, її поява в будь-якому місці модифікованого потоку буде неможливою.

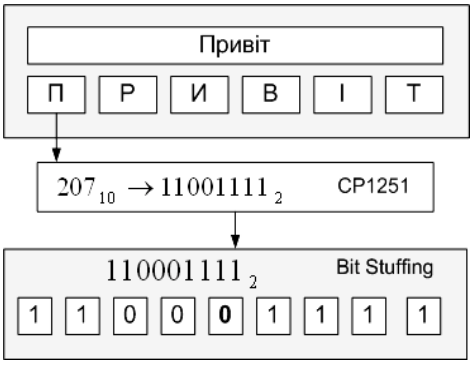

Рис. 3. Вставлення нульового біту в середину октету при Bit Stuffing

Крім FSS у структурі передачі інформації також використовується CRC (Cyclic redundancy check). Під CRC будемо розуміти алгоритм знаходження контрольної суми, призначений для перевірки цілісності даних. CRC є практичним застосуванням завадостійкого кодування, що заснований на певних математичних властивостях циклічного коду.

CRC потрібен для того, щоб унеможливити появу FSS у наступній бітовій послідовності.

Структурна схема отримання повідомлення при цьому має наступний вигляд (рис. 4), де:

t_1' - дійсний час прийому попереднього несучого пакета;

t_2' - дійсний час прийому поточного несучого пакета;

b_0 - бітчасове значення біта 0;

b_1 - бітчасове значення біта 1.

Бітчасові значення b_0 та b_1 повинні знаходиться в інтервалі $[0,\omega)$, і кожен з цих двох b_0 та b_1 бітчасових значень однозначно визначається довільним вибором бітчасового значення протилежного біту (для 0 це буде - 1 і навпаки).

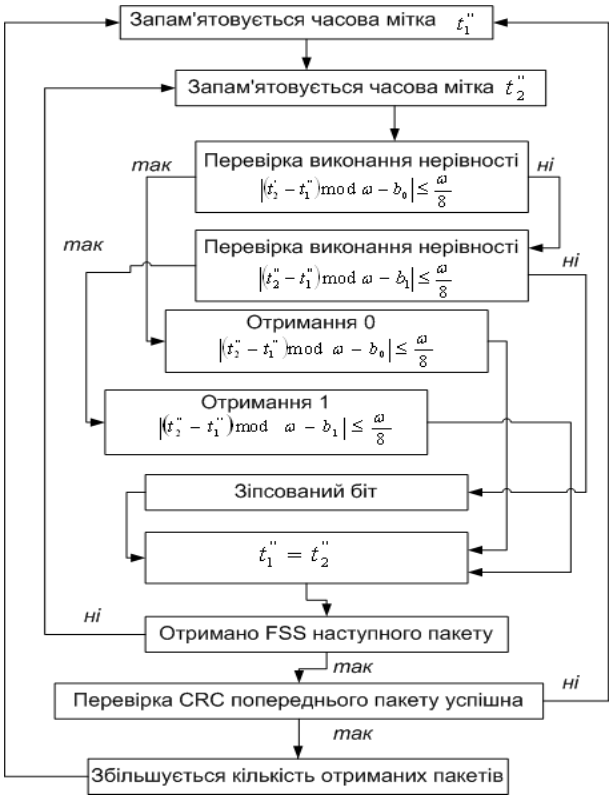

Рис. 4. Структурна схема отримання повідомлення

Основними компонентами, що застосовує розроблена модель є:

- користувачі (сторона передачі і прийому інформації);
- стороннє програмне забезпечення чиї пакети даних піддаватимуться часовій модуляції;
- розроблене програмне забезпечення, що здійснює часову модуляцію пакетів даних;
- мережа.

Розглянемо більш детально кожен з компонентів:

Local Application – програмне забезпечення сторонніх виробників, що відправляє та приймає пакети даних, таким чином, спілкуючись з Remote application.

Remote application - програмне забезпечення сторонніх виробників, що знаходиться на протилежній стороні каналу зв'язку.

Developed software – розроблене програмне забезпечення, що маніпулює та аналізує затримки пакетів на обох сторонах каналу зв'язку з метою встановлення каналу зв'язку у відповідності з запропонованою моделлю.

Developed software складається з основних ключових компонентів: Forwarder (приймач-передавач пакетів), Mediator (посередник при обміну даними між компонентами), Message Codec (кодувач/декодувач повідомлень), PRNG (генератор псевдовипадкових послідовностей), Bit Decoder (декодувач бітів), Bit Encoder (кодувач бітів), Pre-Configurator (формувач початкової конфігурації генератора псевдовипадкових послідовностей (ГПСП)), HASHER – компонент, за допомогою якого здійснюється хешування паролів.

Forwarder (приймач-передавач пакетів) відкриває 2 сокета на прослуховування вхідних UDP пакетів - один з локального застосунка, інший - з віддаленої сторони. Кожен з цих сокетів використовується як для прийому, так і для відправки пакетів відповідній стороні. Відправлені пакети не змінюються після отримання з протилежного сокету. Потік, який обробляє ці повідомлення, повинен мати високий пріоритет для досягнення максимальної точності вбудовування та вимірювання затримок. Але навіть при цьому, маючи на увазі те, що Windows не є ситемой реального часу і, саме тому, не гарантує часову точність при «пробудженні» потоку, для вбудування затримки в вихідні пакети - застосовується 2 етапи очікування.

Перший етап очікування - це «усипляння» потоку на об'єкті таймера очікування , таймаут якого задається цілком менше, ніж необхідно для вбудовування затримки.

Другий етап очікування використовує «холостий цикл» затримки на лічильнику. Згідно з моделлю, вбудування затримки повиненно дозволяти досягати точності до $0{,}1\mu s$. Як на першому етапі очікування, так і на другому передбачається не тільки витікання часу, але і поява пакетів на кожному з сокетів, що дозволяє забезпечити максимально швидку обробку одержуваного пакета.

Mediator (посередник при обміні даними між компонентами) координує дії і відображає призначений для користувача інтерфейс.

Крім того, він розраховує час, на який потрібно затримати пакет для кодування біта, який отримано від Message Codec, на основі значення, яке повертає Bit Encoder, а також часу, що минув з моменту відправлення попереднього несучого пакета.

Message Codec (кодувач/декодувач повідомлень) здійснює формування та розшифування повідомлень. Під час розшифрування повідомлень, Message Codec буферизує послідовно вхідні біти, а виконує їх - Bit Destuffing (операція протилежна Bit Stuffing, що включає в себе перевірку і отримання одиничного біта в кожній дев'ятибітовій послідовності). Як тільки Message Codec зустрічає дев'ять

послідовних одиничних біт, тобто FSS, то виконується перевірка CRC отриманого повідомлення і, якщо CRC послідовність вірна, то збільшується кількість отриманих повідомлень у внутрішньому лічильнику та передається текст повідомлення до Mediator. При формуванні повідомлень на початку Message Codec додає FSS, розраховує CRC повідомлення і додає його в заголовок, а також розраховує кількість успішно відправлених і отриманих пакетів, після чого виконує Bit Stuffing для кожного октету відправленого повідомлення після FSS.

Основною функціональністю PRNG (ГПВП) є повернення значень в інтервалі від 0 до 20000, які в подальшому використовуються як значення 0 біта в мікросекундах.

Bit Decoder (декодувач бітів) відстежує найпершу появу FSS в бітовому потоці, що дозволяє відфільтрувати пакети, які ще не використовувалися для кодування повідомлення відправником. Як тільки така послідовність (FSS) виявлена, Bit Decoder переходить в режим декодування бітів (демодуляції затримок). В цьому режимі він відстежує потрапляння кожної затримки в інтервали толерантності біта 0 та біта 1 і, таким чином, може виявляти пошкодження даних на ранньому етапі декодування повідомлень.

Bit Encoder (кодувач бітів) отримує значення біта і повертає код отриманого біта в мікросекундах на основі значення повернутого PRNG.

Pre-Configurator (формувач початкової конфігурації ГПВП) запитує у користувача IP адресу і порт одержувача, локальний порт, на якому будуть очікуватися вхідні пакети, а також пароль для ініціалізації PRNG (Bit Encoder та Bit Decoder). Перед ініціалізацією PRNG паролі передаються HASHER.

HASHER (компонент, за допомогою якого здійснюється хешування паролів) приймає паролі і, за допомогою використання стандартного криптопровайдера операційної системи Windows, обчислює хеши і повертає їх до Pre-Configurator для подальшої ініціалізації PRNG.

Розглянемо процес відправки одного пакета даних з Local Application до Remote application:

1. Відправлений пакет приймається через Forwarder. Вміст пакету аналізується для виявлення несучого пакету. Решта пакетів передаються без затримки і без урахування самого пакету. А щодо несучих пакетів, Forwarder розрізняє два випадки:

1.1 Якщо це перший несучий пакет, що відправляється у мережу, Forwarder запам'ятовує час його відправлення і передає його в мережу без затримування;

1.2 Наступні несучі пакети затримуються на кількість мікросекунд, що зазначені Mediator у відповідь на повідомлення про появу несучого пакету.

2. Коли Forwarder повідомляє Mediator про відправку несучого пакету, Mediator робить запит до Message Codec про наступний біт повідомлення, що потребує відправки.

3. Message Codec попередньо зберігавший отримане від Mediator повідомлення, формує пакет даних, що складається з bit stuffing FSS, CRC, кількості відправлених та отриманих повідомлень а також текст повідомлення, та віддає до Mediator черговий біт.

4. Для трансформації отриманого біта Mediator звертається до Bit Encoder.

5. Bit Encoder опитує ГПВЧ на предмет поточного значення нульового біту в мікросекундах і на підставі цього, якщо потрібно, обчислює значення одиничного біта (якщо прийшла 1) і передає результат обчислення до Mediator.

6. Mediator, на підставі отриманого від Forwarder часу, що пройшов з моменту відправки останнього несучого пакету, розраховує додаткову затримку і повідомляє Forwarder про те, на який час треба затримати поточний пакет.

7. Прочекавши заданий час, Forwarder відправляє пакет в мережу.

Розглянемо процес прийом пакету що знаходиться на стороні Remote application:

1. Forwarder приймає пакет, який був переданий через Developed software, що знаходиться на стороні Remote application, та запам'ятовує момент його приходу і передає без затримки до Local Application.

2. Forwarder розраховує пройдений час з моменту отримання попереднього несучого пакета і затримку, що розрахована, передає до Mediator.

3. Отримана затримка передається Bit Decoder для розшифровки.

4. Bit Decoder запрошує поточне значення нульового біту у ГПВЧ та порівнює його з отриманой затримкою. На підставі порівняння вирішується, чи отримано нульовий, одиничний або пошкоджений біт.

5. Bit Decoder буферизує біти до того моменту, поки не отримана найперша FSS.

6. Mediator розрізняє два випадки:

6.1 Якщо Bit Decoder ще не виявив першу відправлену FSS, то ніяких дій з розшифрованим бітом не проводиться;

6.2 Якщо Bit Decoder повідомляє, що FSS була отримана раніше, розшифрований біт передається до Message Codec.

7. Message Codec акумулює отриманий біти і, як тільки отримано не пошкоджене повідомлення, передає його до Mediator для виведення на екран.

Висновки

Цільовий аналіз потоку пакетів на предмет вбудованих затримок не дозволить відрізнити їх від

випадковості моменту відправки цих пакетів стороннім додатком – це дозволяє вирішити задачу скритності факту передачі даних між абонентами в мережі.

Виявлення передачі неможливо без знання поточного стану ГПВЧ [6]. FSS - це дев'ять одиниць (біт), але кожній одиниці відповідає випадкова затримка, тому, при спробі виявлення «видно» тільки випадкові моменти появи пакетів в мережі. Те, що ці випадкові моменти кодують саме одиниці, знають тільки відправник і одержувач тому, що тільки їм відомо стан ГПВЧ [7, 8].

Для пакетів, для яких мережевий шум виходить за межі інтервалу толерантності, виникають втрати, які покриваються за рахунок повторної передачі одного і того ж пакету. Виділення інформації прямо пропорційно її вбудові: вимірюються інтервали між послідовними несучими пакетами. Кожен інтервал відповідає одному біту, який декодується відповідно до нерівності

$$\left|(t_2' - t_1') \bmod \omega - b_0\right| \le \omega / 8,$$

де t_2' - момент приходу останнього несучого пакету, t_1' - момент приходу передостаннього несучого пакету, ω - величина часового вікна (20 мс), b_0 - поточне бітчасове значення для біта 0 або біта 1. Той біт, для якого виконується зазначена нерівність і є результат декодування. Якщо воно не виконується ні для одного з них, то біт вважається загубленим. Саме значення b_0 обчислюється на основі результату роботи синхронізованого з відправником ГПВЧ.

Кількість повторних відправок не фіксоване, а визначається підтвердженням доставки з боку отримувача повідомлення. Підтвердження доставки приходить від отримувача повідомлення теж у вигляді затримок.

Згідно розробленої моделі, при побудові каналу зв'язку необхідно враховувати наступне:

- використовувати стабільне підключення до найближчого мережевого шлюзу;
- використовувати захищені канали обміну даними для передачі поділяємого секрету;
- використовувати криптостійкий генератор ПВП;
- проводити аналіз мережі на придатність до вбудови затримок та обирати розмір часового вікна, згідно з проведеним аналізом.

Список літератури

1. Вишневский В.М. Теоретические основы проектирования компьютерных сетей / В.М. Вишневский. – М.: Техносфера, 2003. – 512 с.;

2. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы. – 4-е изд. / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2012. – 943 с.;

3. Остерлох Х. Маршрутизация в IP-сетях. Принципы, протоколы, настройка: пер. с англ. / В. Плешаков. – СПб.: ДиаСофтЮП, 2002. – 512 с.;

4. Горбенко І.Д. Теоретичні основи побудови криптографічних систем абсолютної стійкості / О.А Замула // Системи обробки інформації.- Х.: ХУПС, 2013.-Вип. 4 (111). – С. 101-105;

5. R. A. Wagner, M. J. Fischer. The string to-string correction problem. J. ACM 21 1 (1974). – P. 168—173;

6. Замула О.А. Аналіз і обґрунтування критеріїв і показників ефективності криптографічних генераторів псевдовипадкових чисел / О.А Замула, Д.О. Семченко, Ю.В. Землянко // Системи обробки інформації. – 2014. – № 4 (120). – С. 131-136;

7. Замула А.А. Генераторы псевдослучайных чисел, основанные на дискретном логарифме / А.А. Замула, Д.А. Семченко // Технологический аудит и резервы производства. – 2013. – № 5/1 (13). – С. 28 – 31;

8. Замула А.А. Методы генерации псевдослучайных последовательностей и оценка их свойств / А.А. Замула, Д.А. Семченко // Прикладная радиоэлектроника. – 2012. – № 2 (11). – С. 191-194.

Надійшла до редколегії 3.11.2015

Рецензент: д-р техн. наук, проф. В.А. Краснобаєв, Харківський національний університет імені В.Н. Каразіна, Харків.

РАЗРАБОТКА И ПРИМЕНЕНИЕ МОДЕЛИ ЗАЩИЩЕННОГО КАНАЛА ПЕРЕДАЧИ ДАННЫХ ЗА СЧЕТ УПРАВЛЕНИЯ НЕФУНКЦИОНАЛЬНЫЕ СВОЙСТВА СЕТЕВЫХ ПАКЕТОВ

Д.А. Семченко, А.А. Замула

Представлена модель защищенного канала связи, которая основана на управлении нефункциональными свойствами сетевых пакетов. Приведены минимально-допустимые требования относительно использования разработанной модели, а так же структура модели и каждого ее компонента. Указанны особенности разработанной модели и сформулированы предложения по использованию модели в современных телекоммуникационных системах.

Ключевые слова: *модель, сообщение, канал связи, временные задержки, пакет данных, битвременные значения.*

DEVELOPMENT AND USAGE MODELS FOR PROTECTED COMMUNICATION CHANNEL BASED ON THE NON-FUNCTIONAL NETWORK PACKETS PROPERTIES

D.A. Semchenko, A.A. Zamula

Presented model of a secure communication channel, which builds on the management of non-functional properties of network packets. Also presented the minimally acceptable requirements for the usage of the developed model, structure of the model and each of its components. Listed features of the developed model and formed proposals for the usage of this model in modern telecommunication systems.

Keywords: *model, message, time delay, data package, bit values.*