УДК 004.9

О. В. Бакай
Національний університет “Львівська політехніка”,
кафедра захисту інформації

ОСОБЛИВОСТІ ПОБУДОВИ СИСТЕМИ ТА ЗАСТОСУВАННЯ КРИПТОГРАФІЧНОГО АЛГОРИТМУ AES

Під час проведення досліджень створено дві програмні реалізації криптографічного алгоритму AES із засобами візуалізації, повноцінною реалізацією алгоритмів шифрування та дешифрування за стандартом AES, засобами протоколювання раундових перетворень шифру з можливостями подальшого детального ознайомлення із проміжними результатами. Також було використано можливості інтерфейсу програмування додатків для генерування ключів. Застосування об'єктно-орієнтованого програмування допомогло осмислити задачу, яка постає під час розроблення програмних модулів, а також визначити шлях до її розв’язання зрозумілішим, а отже, й ефективнішим способом. Зокрема, розглянутий алгоритм AES можна використати замість Triple DES, щоб підвищити рівень інформаційної безпеки здійснення трансакцій з використанням карткових платіжних систем.

Ключові слова: шифрування, криптопровайдер, NIST, ECB, алгоритми DES, Triple DES, AES.

CHARACTERISTICS OF SYSTEM CONSTRUCTION AND IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHM AES

Within investigation of AES-standard usage two software modules were implemented. These realizations include visualization tools, full implementation of algorithms for encryption and decryption according to AES standard, means for detailed logging of encryption changes to have possibilities see intermediate results data. Also application programming interface capabilities has been used to generate keys. The use of object-oriented programming allows understand the problem deeper as well as get the best way to solve it more clearly and therefore in more effective way. In particular, the AES algorithm could be used instead of Triple DES for information security of payment card systems transactions.

Key words: encryption, cryptoprovider, NIST, ECB, crypto algorithms AES, Triple DES, AES.

Вступ

XXI століття – епоха інформатики та інформатизації, основною цінністю є інформація. Технологія дає можливість передавати і зберігати все більші обсяги інформації. Це благо має й зворотний бік. Інформація стає значно вразливішою з різних причин:

- зростання обсягів збережених і переданих даних;
- розширення кола користувачів, що мають доступ до інформаційних систем;
- ускладнення інформаційних систем.

Напрям розроблення системи стандартів із захисту даних, що зберігаються і обробляються на ЕОМ, від несанкціонованого доступу визнано однією із найпріоритетніших областей. Одним із найвідоміших дослідницьких центрів в ті часи була наукова лабораторія фірми IBM, в якій зародився стандарт шифрування DES (Data Encryption Standard).

Донедавна DES використовували в США, за незначними винятками, практично скрізь: в урядовому зв'язку, електронних банківських переказах, цивільних супутникових комунікаціях і паролях комп'ютерних систем.

DES критикують за кілька недоліків, зокрема за дуже маленьку довжину ключа – всього 56 розрядів. Дослідники зазначили, що такої довжини ключа буде цілком достатньо на 10–15 років, і вони не помилилися. Про всяк випадок було вирішено переглядати стандарт кожні п'ять років.

Такі перегляди виконано в 1983, 1988 і 1993 рр., тоді стандарт був залишений без змін. Проте незабаром слабкість шифру стала очевидною і для збільшення стійкості фахівці рекомендували, використовуючи його, виконувати шифрування два або три рази з різними ключами. У 1998 р., коли всім стало остаточно зрозуміло, що стандарт шифрування треба замінити, Національний інститут стандартів і технологій (NIST) опублікував запит, в якому описувався передбачуваний "Вдосконалений стандарт шифрування" (Advanced Encryption Standard – AES), що повинен прийти на зміну DES.

Аналіз літературних джерел

Для того, щоб алгоритм AES [1–3] став гідною заміною DES, його архітектура повинна задовольняти декілька критеріїв: надавати високий ступінь захисту, мати просту структуру та високу продуктивність.

Очевидно, що захист є найвищим пріоритетом для алгоритму AES. Вже на рівні внутрішньої архітектури він повинен мати надійність, достатню для того, щоб протистояти майбутнім спробам його зламу.

Разом з тим структура алгоритму, на противагу традиційним поглядам, повинна бути настільки простою, щоб гарантувати ефективну процедуру шифрування.

Наступною у списку вимог до AES є висока продуктивність. Якісна продуктивність алгоритму припускає високу швидкість роботи під час шифрування і дешифрування, а також реалізації графіка ключа. Щоб бути затвердженим як стандарт, алгоритм повинен:

- Реалізувати шифрування приватним ключем.
- Являти собою блочний шифр.
- Працювати з 128-розрядними блоками даних і ключами трьох розмірів (128, 192 і 256 розрядів).

Додатково кандидатам рекомендувалося:

- Використовувати операції, які легко реалізуються як апаратно, так і програмно.
- Орієнтуватися на 32-розрядні процесори.
- Не ускладнювати без необхідності структуру шифру [4].

Є багато областей, в яких AES зараз комерційно використовується. VPN програмне забезпечення найвищого класу містить реалізацію AES, зокрема пропозицій від Checkpoint, Cisco і Symantec. AES тепер часто застосовують у мережевих пристроях. Voice Over IP виробники використовують AES для безпеки телефону. Продавці тепер використовують AES для убезпечення управління технологічними процесами (SCADA) систем. AES навіть додано в загальний файл стиснення програм, таких як WinZip.

Для організованої побудови програмних засобів захисту інформації (ЗЗІ) найхарактерніша тенденція розроблення комплексних програм, що виконують багато захисних функцій, причому найчастіше до цих функцій входить розпізнавання користувачів, розмежування доступу до масивів даних, заборона доступу до деяких областей оперативної пам'яті тощо. Переваги таких програм очевидні: кожна з них забезпечує розв'язання низки важливих задач захисту.

З описаних недоліків та переваг випливають такі вимоги до формування програмних ЗЗІ: функціональна повнота, гнучкість і уніфікованість використання [5].

Найповніше вимоги гнучкості та уніфікованості задовольняють такі сукупності принципів: наскрізна модульна будова, повна структуризація, представлення на машиннонезалежній мові.

Принцип наскрізної модульної побудови полягає в тому, що кожна з програм будь-якого рівня та об'єму має подаватися у вигляді системи можливих модулів, причому кожний модуль будь-якого рівня має бути повністю автономним і мати стандартні вхід та вихід, що забезпечує компонування з будь-якими іншими модулями. Очевидно, що ці умови можуть бути виконані, якщо

програмне забезпечення розроблятиметься за принципом “згори донизу”, тобто відповідно до принципу повної структуризації.

Представлення на машиннонезалежній мові передбачає таке подання програмних модулів, щоб їх із мінімальними зусиллями можна було ввести до складу програмного забезпечення будь-якої АС. Сучасні алгоритмічні мови високого рівня повністю відповідають цим вимогам (такі мови часто є трансплатформними та підтримуються середовищами MS CryptoAPI 2.0 та.NET Framework).

Використання CryptoAPI

CryptoAPI – інтерфейс програмування додатків, який забезпечує розробників стандартним набором функцій для роботи з криптопровайдером. Входить до складу операційних систем Microsoft. Більшість функцій CryptoAPI підтримуються, починаючи з Windows 2000.

Криптопровайдер – це зазвичай бібліотека, що реалізує певний набір криптографічних алгоритмів і забезпечує роботу із ними. Існує близько семи стандартних провайдерів, попередньо встановлених у системі.

Кожен криптопровайдер належить до певного типу. Це дає змогу, перебравши всі встановлені в АС провайдери, вибрати ті, які підтримують потрібні алгоритми.

Криптопровайдер підтримує захищені області, що називають контейнерами ключів. Контейнери дають можливість додаткам зберігати і використовувати надалі згенеровані один раз ключі, забезпечуючи захист самих ключів від компрометації та модифікування.

Більша частина криптографічних класів (не абстрактних).NET базується на криптопровайдерах CryptoAPI. Однак є і винятки (наприклад SHA256Managed). У принципі, ієрархія класів.NET дозволяє абстрагуватися від конкретної реалізації алгоритму, що може забезпечити простий перехід на не прив’язані до CryptoAPI класи в майбутньому.

.NETFramework містить набір криптографічних сервісів, що розширюють аналогічні сервіси Windows через CryptoAPI. Простір імен System.Security.Cryptography відкриває програмний доступ до різноманітних криптографічних сервісів, за допомогою яких додатки можуть шифрувати та дешифрувати дані, забезпечувати їх цілісність, а також обробляти цифрові підписи та сертифікати.

На найвищому рівні простір імен Cryptography можна розділити на чотири основні частини, як показано в табл. 1.

Таблиця 1

Основні елементи простору імен Cryptography

Елемент	Опис
Алгоритми шифрування	Набір класів, що застосовуються для симетричного та асиметричного шифрування, а також хешування
Допоміжні класи	Класи, що забезпечують генерацію випадкових чисел, виконання перетворень, взаємодію із бібліотеками CryptoAPI і саме шифрування на основі потокової моделі
Сертифікати X.509	Класи, визначені в просторі імен System.Security.Cryptography. X509 Certificates, які представляють цифрові сертифікати
Цифрові підписи XML	Класи, що визначають в просторі імен System.Cryptography.Xml і представляють цифрові підписи в XML-документах

Основне призначення цього простору – надавати класи з алгоритмами таких операцій, як шифрування і створення хешів. Ці алгоритми реалізуються на основі розширюваного шаблону, що містить два рівні наслідування.

На вершині ієрархії розташовується абстрактний базовий клас, ім’я якого відповідає типу алгоритму. Від такого класу наслідує абстрактний клас другого рівня, що надає відкритий інтерфейс для використання цього алгоритму. Наприклад, SHA1 (Secure Hash Algorithm) являє собою похідний від HashAlgorithm клас і містить методи і властивості, специфічні для алгоритму SHA1. Сама реалізація алгоритму є похідною від класу другого рівня; саме її екземпляр

створюється і використовується клієнтським додатком. На цьому рівні реалізація може бути керованою, некерованою, а також і тією, й іншою.

Створюючи екземпляр одного з конкретних класів, початкові конструктори завжди записують в параметри об’єкта обґрунтовані та безпечні значення, якщо це можливо. Так, алгоритми асиметричного шифрування, що спираються на криптографію з відкритим ключем, генерують випадкову пару ключів, а алгоритми симетричного шифрування – випадковий ключ і вектор ініціалізації; при цьому вони автоматично налаштовують властивості. І навіть більше, алгоритми другого типу за замовчуванням намагаються використовувати “стійкі” значення.

Реалізація системи Rijndael засобами мови Java

Розроблена у ході дослідження програма спроектована як багатофункціональна система візуалізації, аналізу і, власне, реалізації шифрування даних з використанням криптографічного алгоритму AES. Основними компонентами системи є:

• Підсистема шифрування за стандартом AES, що містить модулі для реалізації шифрування і дешифрування з підтримкою режиму електронної кодової книги (Electronic codebook, ECB).

• Підсистема візуалізації, яка розкриває суть алгоритму і принципів перетворень, реалізованих у ньому, за допомогою демонстрації процесу шифрування та відповідного графічного матеріалу.

Система володіє всіма необхідними засобами для виконання операцій блокового перетворення 128, 192, 256-бітних ключів та блоків даних незалежно один від одного, і, відповідно, 10, 12 і 14-раундових перетворень. Обмеження на розмір даних, що шифруються, залежать тільки від кількості адресованої віртуальної пам'яті, доступної системі [2, 3].

Основними інформаційними блоками в роботі програми є стани відкритого та шифрованого тексту. На основі введеного користувачем ключа система виконує планування підключів і генерує ключові послідовності для шифрування та дешифрування.

Реалізацію цієї системи вирішено виконувати засобами об'єктно-орієнтованого програмування, на основі програми у вигляді сукупності об’єктів, кожний з яких є реалізацією визначеного типу, що використовує механізм пересилання повідомлень та класи, організовані в ієрархію успадкування.

Інструментом для створення програми вибрано мову програмування Java. Програми на Java транслюються у байтовий код, що виконує віртуальна машина – програма, яка обробляє байтовий код і передає інструкції обладнанню як інтерпретатор, однак байтовий код, на відміну від тексту, обробляється значно швидше.

У процесі шифрування використовується режим електронної кодової книги (ECB), також відомий як метод простої заміни [6]. У загальному випадку його рекомендують як найбільш простий і швидкий режим роботи блочного шифру.

Режим ECB – найпростіший режим шифрування. Всі блоки відкритого тексту шифруються незалежно один від одного. Це важливо для шифрованих файлів з довільним доступом, наприклад, файлів баз даних. Якщо база даних зашифрована в режимі ECB, будь-який запис може бути доданий, вилучений, зашифрований або розшифрований незалежно від будь-якого іншого запису (за умови, що кожен запис складається із цілої кількості блоків шифрування). Окрім того, обробка може бути паралельною: якщо використовуються декілька шифрувальних процесорів, вони можуть шифрувати або розшифровувати різні блоки незалежно один від одного.

Серед функцій системи варто також відзначити:

• Наявність інструментарію візуалізації алгоритму AES, який розкриває суть алгоритму і принципів перетворень, реалізованих у ньому: вигляд діалогового вікна програми показано на рис. 1.

• Наявність журналу раундових перетворень (рис. 2), який містить докладний звіт про трансформацію даних кожного шифрованого і дешифрованого блока на всіх етапах раундових перетворень.

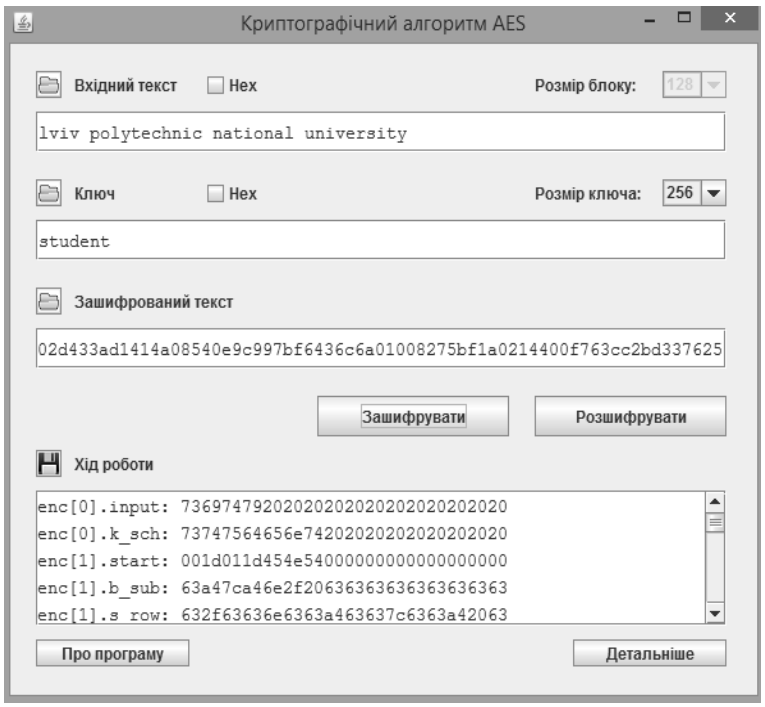

Рис. 1. Інтерфейс програмного модуля в процесі зашифровування

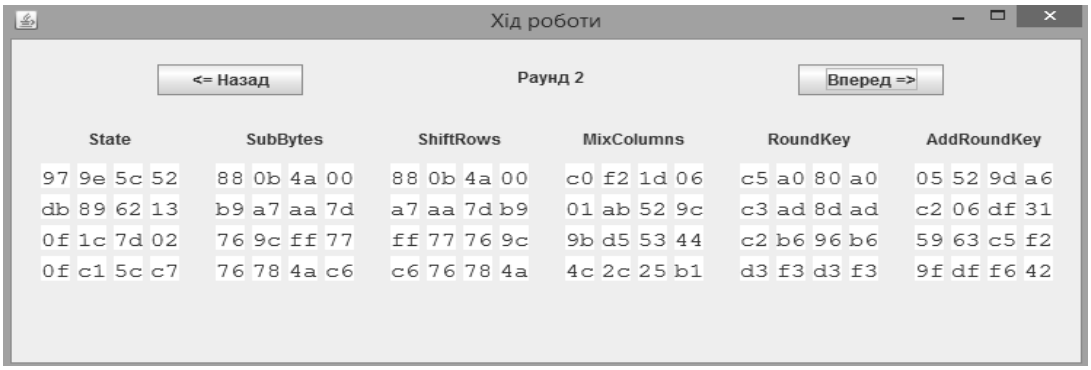

Рис. 2. Демонстрація другого раунду зашифровування

Переваги алгоритму AES, пов'язані з реалізацією:

• Rijndael може виконуватися швидше, ніж ранні блокові алгоритми шифрування. Забезпечено оптимізацію між розміром таблиці та швидкістю виконання.

• Rijndael можна реалізувати у вигляді коду, використовуючи невелику ОЗП (RAM) і маючи невелику кількість циклів. Виконано оптимізацію розміру ROM і швидкості виконання.

• Перетворення раунду допускає паралельне виконання, що є важливою перевагою для сучасних багатоядерних процесорів і спеціалізованої апаратури.

• Алгоритм шифрування не використовує арифметичні операції, тому тип архітектури процесора практично не має значення.

Можливості розширення:

• Розробка дозволяє специфікувати варіанти довжини блока і довжини ключа в діапазоні від 128 до 256 бітів з кроком у 32 біти.

• Хоча кількість раундів Rijndael зафіксована в цій специфікації, у разі виникнення проблем з безпекою він може модифікуватися і мати таку кількість раундів, як параметр.

Приклад реалізації основних симетричних алгоритмів мовою C# в.NET Framework

Симетричні шифри DES і 3DES (TDES) – одні з небагатьох симетричних шифрів, що надаються стандартними криптопровайдерами CryptoAPI. Оскільки DES і 3DES – практично той самий алгоритм, то можна обмежитись прикладом використання 3DES.

Однак існує відмінність: для використання алгоритму 3DES необхідний *Enhanced* провайдер, коли для DES достатньо *Base*. Загалом, DES нині вже не є стійким алгоритмом, тому його бажано використовувати лише в тих системах, в яких надійність шифрування не є критичною.

Алгоритм 3DES використовує різні ключі DES для кожної зі своїх ітерацій. Тому розмір його ключа дорівнює потрійному розміру ключа DES: 192=64*3 біти. Реальний розмір ключа 3DES 168=56*3 бітів, оскільки в DES один байт ключа є контрольним для основних семи [7].

Для наочної демонстрації шифрування/розшифрування даних у разі передавання їх у незахищеному середовищі розглянемо програмний модуль, вигляд якого наведено на рис. 3.

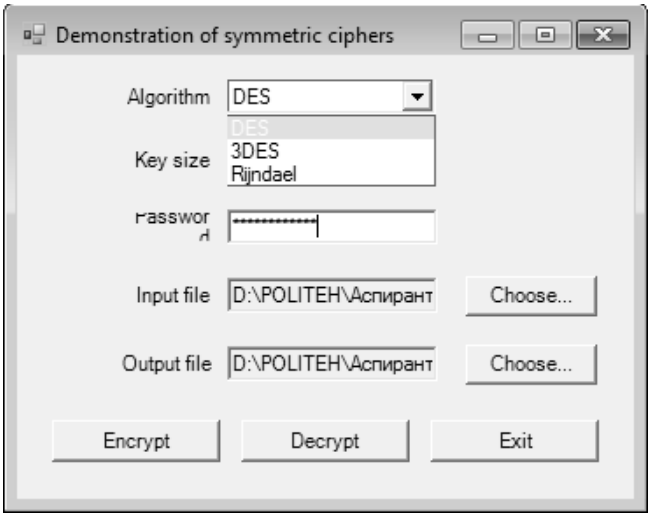

Рис. 3. Інтерфейс програмного модуля

Як бачимо, перше поле з випадним меню дає змогу вибрати один з представлених симетричних алгоритмів шифрування. Окрім DES і 3DES, реалізовано також алгоритм AES, який в.NET представлений класом Rijndael.

Цей алгоритм підтримує ключі довжиною 128, 192 та 256 біт.

У полях “Input file/Output file” вибираються адреси файла, в якому міститься початковий текст, і файла, в який слід записати зашифрований текст. У поле “Password” вводимо пароль довільної довжини, на основі якого генеруються ключ K1 і вектор ініціалізації.

Нижче наведено відкритий текст і отримано криптограму цього тексту, зашифрованого алгоритмом Triple DES, довжина ключа 192 біти з паролем “Rhbgnjpf[bcn” (рис. 4, 5).

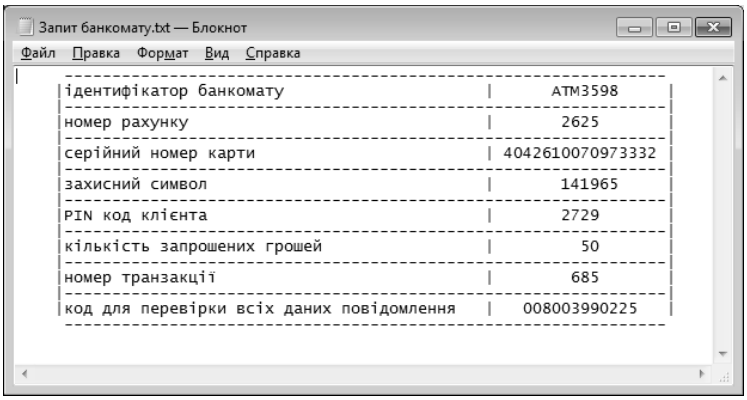

Рис. 4. Відкритий текст

Рис. 5. Отримана шифрограма

Розшифрування проводять аналогічно, вказавши відповідні адреси файлів і з використанням того самого пароля. Якщо пароль змінюється, відповідно змінюються і згенеровані на його основі ключ і вектори ініціалізації, відтак розшифрування унеможливлюється. Програмний модуль реагує на це вікном із повідомленням про помилку і якщо при цьому натиснути “continue”, то файл, в який мали б були записані розшифровані дані, залишиться пустим.

Отже, підсумовуючи усе вищеописане, можна зробити висновок, що цей програмний модуль має такі переваги:

- Варіативність – проявляється у можливості вибрати алгоритм шифрування та ступінь його стійкості (довжину ключа) безпосередньо в процесі шифрування кожного файла.
- Гнучкість – модуль написано об’єктно-орієнтованою мовою високого рівня C# і реалізовано на платформі.NET Framework, що уможливлює широке застосування у різних системах як фірми Microsoft, так і у Unix-подібних середовищах.
- Простота використання – постає у вигляді доступного простого інтерфейсу [8].

Оскільки використані у модулі алгоритми є міжнародними стандартами шифрування, то це означає, що такий модуль можна використовувати як засіб забезпечення безпеки даних під час зберігання їх та передавання по незахищеній мережі, дотримуючись вимог стандарту PCI DSS.

Висновки

У ході дослідження розглянуто дві доволі різнопланові реалізації симетричних алгоритмів шифрування на прикладі системи AES. Реалізована мовою Java програма дає можливість глибше та детальніше розібратися в структурі та принципах процесу зашифровування даних, відстежити процеси підстановок та перестановок упродовж кожного раунду. Натомість програмний модуль, створений у середовищі.NET Framework, більше відповідає сучасним вимогам до програмних засобів захисту інформації, зокрема і в банківській сфері.

Обидва підходи мають переваги під час як реалізації, так і застосування. Перший варіант доцільніше використовувати з навчальною метою, оскільки він дає змогу створити уявлення про принципи реалізації сучасних криптоалгоритмів із застосуванням об’єктно-орієнтованого підходу. В другому ж варіанті виконання маємо повноцінний віконний програмний модуль із прозорим та інтуїтивно зрозумілим інтерфейсом, створений із використанням усіх переваг вже існуючих та вбудованих в більшості операційних систем бібліотек методів криптозахисту.

1. Ruohonen K. Mathematical Cryptology. – Tampere: Tampere University of Technology. – 2010. – 136 p. 2. Панасенко С. П. Алгоритмы шифрования: спец. справочник / С. П. Панасенко. – СПб.: БХВ-Петербург, 2009. – 576 с. 3. Зензин О. С., Иванов М. А. Стандарт криптографической защиты – AES. Конечные поля / под ред. М. А. Иванова. – М.: КУДИЦ-ОБРАЗ, 2002. – 176 с. 4. Daemen J., Rijmen V. Design of Rijndael. AES – The Advanced Encryption Standard. – Berlin: Springer–Verlag. – 2002. – 238 p. 5. Хорошко В., Чекатков А. Методы и средства защиты информации. – К.: Юниор, 2003. – 504 с. 6. Чмора А. П. Современная прикладная криптография. – 2-е изд. – Гелиос АРВ, 2002.

– 256 с. 7. Бакай О. В., Брич Т. Б., Лах Ю. В. Технології захисту банківських міжнародних платіжних карток // Вісник Національного університету "Львівська політехніка": Автоматика, вимірювання та керування. – 2012. – № 741. – С. 184–187. 8. Dudykevych V., Bakay O., Lakh Y. Investigation of Payment Cards Systems Information Security Control // Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), Volume 2, (September 12–14, 2013, Berlin, Germany) P. 651–654.

УДК 616.71

Л. О. Березко, С. Є. Соколов*
Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин,
*кафедра теоретичної радіотехніки та радіовимірювання

ОСОБЛИВОСТІ ПРОЕКТУВАННЯ ЕЛЕКТРОННОЇ БІОМЕДИЧНОЇ АПАРАТУРИ

Проаналізовано побудову та особливості електронної біомедичної апаратури. Запропоновано основну структуру біотехнічної системи. Розглянуто перспективи використання мікропроцесорних засобів для розроблення біомедичної апаратури. Сформульовано методологічні рекомендації стосовно процесу розроблення нової біомедичної апаратури.

Ключові слова: біотехнічні системи, комп'ютерні технології.

FEATURES OF ELECTRONIC BIOMEDICAL EQUIPMENT DESIGN

The analysis of electronic biomedical equipment design and features was performed. The main structures of biotechnological systems' is proposed. The prospects for the use of microprocessor means during biomedical equipment development are considered. The methodological recommendations on the biomedical equipment development process were made.

Key words : biotechnical system, computer technology.

Вступ

Розвиток біології та медицини потребує отримання об'єктивних інструментальних показників життєдіяльності біологічних об'єктів (БО). Це можливо тільки за умови розроблення та впровадження у практику нової спеціалізованої біомедичної електронної апаратури (БМЕА). У зв'язку зі специфікою досліджень біологічних об'єктів (БО) необхідні для цього прилади та системи під час проектування та розроблення треба розглядати як елементи спільної з БО біотехнічної системи (БТС) [2, 11]. Такий системний підхід враховує взаємний вплив БО та БМЕА, що дає змогу підвищити ефективність останньої [6, 10]. Незважаючи на значні успіхи світового медичного та біологічного приладобудування, різноплановість та комплексний характер задач проектування надає їм специфічної складності [4, 17].

Один з найперспективніших напрямів розроблення сучасної БМЕА ґрунтується на вимірюванні електричного імпедансу (або його складових) біологічної тканини [2, 5, 8].