Якименко І.З.

Лекції

з дисципліни:

Дослідження та проектування систем захисту інформації

Лекція 1.

Теоретичні засади дослідження та проектування систем захисту інформації.

1.Принципи організації захисту інформації.

2. Концепція національної безпеки України, концепція інформаційної безпеки України, доктрина інформаційної безпеки України.

3.Основні поняття, терміни та визначення.

1. Інформаційна безпека - це стан захищеності інформаційного середовища, захист інформації являє собою діяльність щодо запобігання витоку інформації, що захищається, несанкціонованих і ненавмисних впливів на інформацію, що захищається, тобто процес, спрямований на досягнення цього стану. Метою реалізації інформаційної безпеки будь-якого об'єкта є побудова системи забезпечення інформаційної безпеки даного об'єкту [4]. Для побудови та ефективної експлуатації СЗІБ (система забезпечення інформаційної безпеки) необхідно:

- виявити вимоги захисту інформації, специфічні для даного об'єкта захисту;
- врахувати вимоги національного та міжнародного законодавства;
- використовувати напрацьовані практики (стандарти, методології) побудови подібних СЗІБ;
- визначити підрозділи, відповідальні за реалізацію та підтримку СЗІБ;
- рас проділити між підрозділами області відповідальності у здійсненні вимог СЗІБ;
- на базі управління ризиками інформаційної безпеки визначити загальні положення, технічні та організаційні вимоги, складові політики інформаційної безпеки об'єкта захисту;
- реалізувати вимоги політики інформаційної безпеки, впровадивши відповідні програмно-технічні засоби і способи захисту інформації;
- реалізувати систему менеджменту (управління) інформаційної безпеки (СМІБ);
- використовуючи систему управління організувати регулярний контроль ефективності СЗІБ і при необхідності перегляд і коригування СЗІБ .

Під системою безпеки будемо розуміти організовану сукупність спеціальних органів, служб, засобів, методів і заходів, що забезпечують захист життєво важливих інтересів особистості, підприємства і держави від внутрішніх і зовнішніх загроз (Рис. 1.) [2].

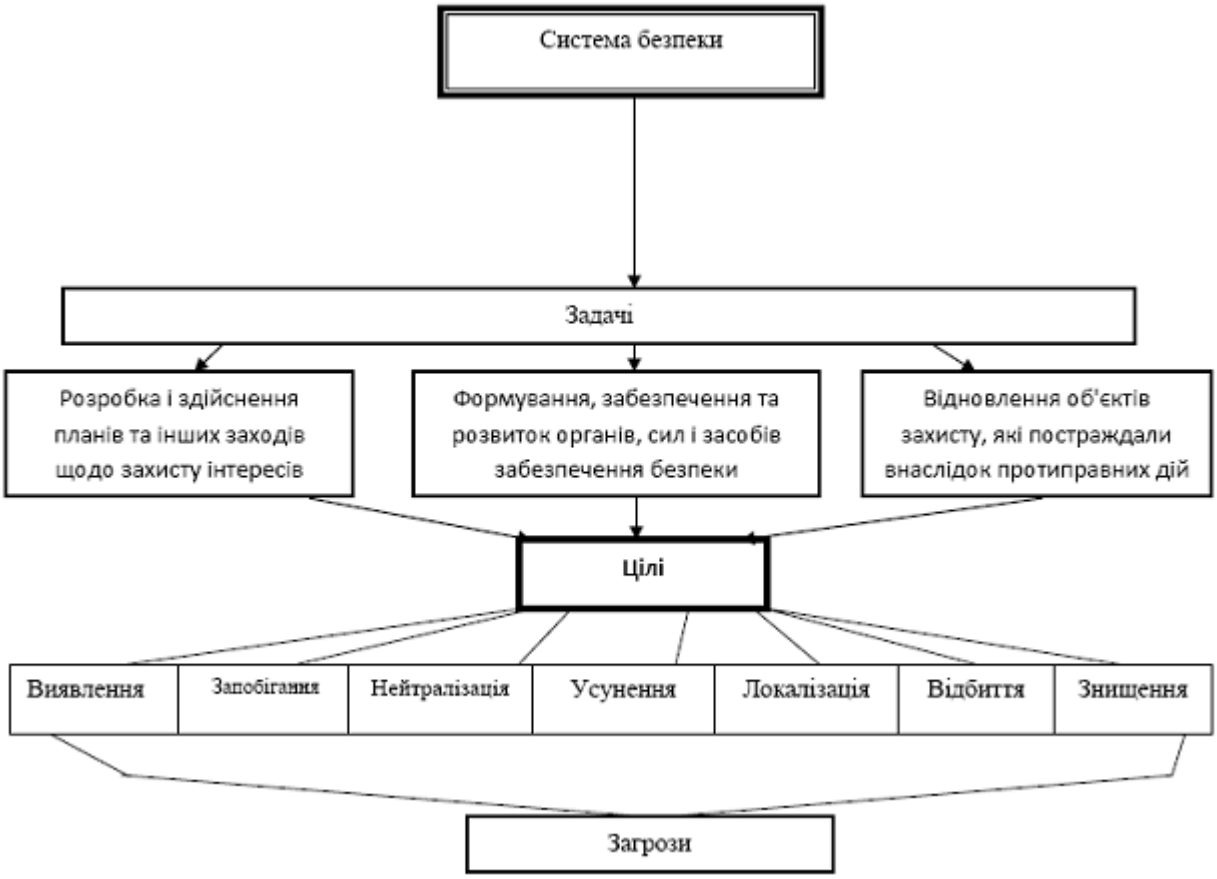


Рис. 1. Функціональна система інформаційної безпеки

Розуміючи інформаційну безпеку як «стан захищеності інформаційного середовища суспільства, що забезпечує її формування, використання і розвиток в інтересах громадян, організацій», правомірно визначити загрози безпеки інформації, джерела цих загроз, способи їх реалізації та мети, а також інші умови і дії, що порушують безпеку . При цьому, природно, слід розглядати і заходи захисту інформації від неправомірних дій, що призводять до нанесення збитку [4].

Загроза - сукупність факторів та умов, що виникають в процесі взаємодії об'єкта безпеки з іншими об'єктами, а також складових його компонентів між собою і здатних чинити на нього негативний вплив. Вона виступає в якості можливості вирішення протиріччя у взаємодії об'єкта безпеки з іншими об'єктами, компонентів об'єкта безпеки, що у стадії дисгармонії чи конфлікту, шляхом насильницької зміни в бік погіршення властивостей об'єкта безпеки, або його компонентів, тобто шляхом нанесення шкоди.

Між загрозою і небезпекою нанесення шкоди завжди існують відносини заподіяння, які визначаються як обумовлена сутністю взаємодіючих об'єктів, елементів системи, зв'язок між явищами, при якій одне явище, зване причиною, за наявності певних умов неминуче породжує, викликає до життя інше явище, зване слідством. Загроза завжди породжує небезпеку. Небезпека може бути визначена як стан, в якому знаходиться об'єкт безпеки внаслідок появи загрози. Відмінність між ними полягає в тому, що небезпека є властивістю об'єкта безпеки, а загроза - властивістю об'єкта взаємодії або знаходяться у взаємодії елементів об'єкта безпеки, виступаючих як джерело загроз. Загроза знаходиться

у відношенні заподіяння не тільки з небезпекою, але і з очікуваним шкодою - наслідками негативної зміни умов існування, які необхідно подолати для відновлення необхідних умов - в тому сенсі, що очікуваний шкоду визначає величину небезпеки [3] .

Загрози інформаційній безпеці - це можливі дії або події, які можуть вести до порушень ІБ. Види загроз інформаційній безпеці дуже різноманітні і мають безліч класифікацій (наведених в рис. 2):

Рис. 2. Види загроз інформаційної безпеки

Відповідно до наведеної вище класифікації загроз за видом об'єкта впливу вони поділяються на загрози власне інформації, загрози персоналу об'єкта та загрози діяльності щодо забезпечення інформаційної безпеки об'єкта. При більш детальному розгляді загроз інформації, їх можна поділити на загрози носіям конфіденційної інформації, місцям їх розміщення (розташування), каналам передачі (системам інформаційного обміну), а також інформації, що зберігається в документованому (електронному) вигляді на різних носіях. За характером порушення, як один із варіантів класифікації, зображено на рис. 3.

Таким чином, можна зробити висновок про те, що дія загроз інформаційній безпеці об'єкта направлено на створення можливих каналів витоку інформації, що захищається (передумов до її витоку) і безпосередньо на витік інформації.

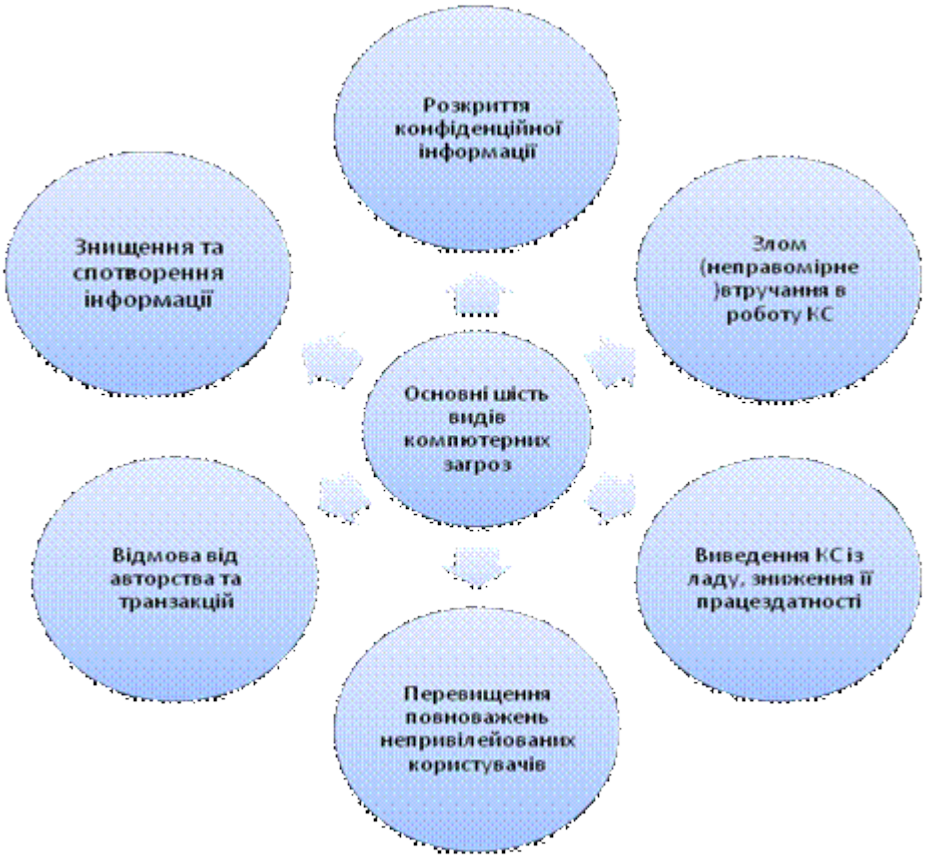


Рис. 3. Шість основних загроз інформаційній безпеці (класифікація за характером порушення)

Одне з ключових понять в оцінці ефективності прояви загроз об'єкту інформаційної безпеки - збиток, що наноситься цьому об'єкту (підприємству) в результаті впливу загроз. За своєю суттю будь-який збиток, його визначення та оцінка мають яскраво виражену економічну основу. Не є винятком і збиток, що наноситься інформаційній безпеці об'єкта (підприємства).

З позиції економічного підходу, загальний збиток інформаційної безпеки підприємства складається з двох складових частин: прямого і непрямого

збитку. Прямий збиток інформаційної безпеки підприємства виникає внаслідок витоку конфіденційної інформації. Непрямий збиток - втрати, які несе підприємство у зв'язку з обмеженнями на поширення інформації, в установленому порядку віднесеної до категорії конфіденційної. Опис збитку, що наноситься підприємству в результаті витоку конфіденційної інформації, грунтується на його кількісних і якісних показниках, які базуються на одному з принципів засекречування інформації (віднесення її до категорії конфіденційної) - принципі обгрунтованості. Він полягає у встановленні (шляхом експертних оцінок) доцільності засекречування конкретних відомостей, а також ймовірних наслідків цих дій, з урахуванням розв'язуваних підприємством задач і поставлених цілей.
Введення обмежень на поширення інформації (у зв'язку з її засекречуванням або віднесенням до категорії конфіденційної) призводить і до позитивних, і до негативних наслідків. До основних позитивних наслідків слід віднести запобігання можливого прямого збитку інформаційної безпеки підприємства через витік інформації, що захищається. Негативні наслідки пов'язані з наявністю (ймовірним зростанням) непрямого збитку або витрат у вигляді витрат на захист інформації та величини упущеної вигоди, яка може бути отримана при її відкритому розповсюдженні.

Загальний збиток безпеки підприємства від витоку конфіденційної інформації визначають наступним чином. Проводять класифікацію всіх наявних на підприємстві відомостей за ступенем їх важливості. З цією метою методом експертної оцінки з залученням фахівців структурних підрозділів підприємства, що беруть участь у виконанні робіт з різних напрямків його діяльності, розробляють єдину шкалу відомостей, що містять конфіденційну інформацію - так званий рейтинг важливості інформації. У рейтингу відбиваються всі відомості, включені до переліків інформації, що підлягає захисту[3].

Методичною основою для розробки такого рейтингу служить метод експертного аналізу в сукупності з методом об'єктивного кількісного оцінювання. На основі рейтингу важливості інформації зіставляють (співвідносять) включені до нього відомості з кількісними показниками можливого збитку, що визначається розрахунковим або експертним шляхом.

При розробці необхідних, засобів, методів і заходів, що забезпечують захист інформації, необхідно враховувати велику кількість різних факторів.

Інформація, будучи предметом захисту, може бути представлена на різних технічних носіях. Її носіями можуть бути люди з числа користувачів і обслуговуючого персоналу. Інформація може піддаватися обробці в комп'ютерних системах, передаватися по каналах зв'язку і відображатися різними пристроями. Вона може розрізнятися за своєю цінністю. Об'єктами, що підлягають захисту, де може перебувати інформація, є не тільки комп'ютери і канали зв'язку, але й приміщення, будівлі та прилегла територія. Істотно різнитися може кваліфікація порушників, а також використовувані способи і канали несанкціонованого доступу до інформації. Таким чином, основними принципами забезпечення інформаційної безпеки є наступні[5]:

Системності.

Комплексності.

Безперервності захисту.

Розумної достатності.

Гнучкості управління і застосування.

Відкритості алгоритмів і механізмів захисту.

Простоти застосування захисних заходів і засобів.

За способами здійснення всі заходи забезпечення безпеки комп'ютерних систем підрозділяють на:

правові (законодавчі);

морально-етичні;

організаційно-адміністративні;

фізичні;

апаратно-програмні.

До *правових заходів* захисту інформації належать діючі в країні закони, укази, положення, інструкції та інші нормативні акти, які регламентують правила поводження з інформацією обмеженого використання і відповідальності за їх порушення. Цим вони перешкоджають несанкціонованому використанню інформації і є стримуючим фактором для потенційних порушників[5].

До *морально-етичним* заходам протидії відносяться всілякі норми поведінки, які традиційно склалися або складаються в суспільстві у міру поширення комп'ютерів в країні. Ці норми бувають як неписаними (загальновизнані норми чесності, патріотизму і т.д.), так і оформленими в якийсь звід правил чи приписів.

Організаційно-адміністративні заходи захисту регламентують процеси функціонування ІС(інформаційних систем); використання ресурсів ІС; діяльність персоналу інформаційної служби на підприємтсві; порядок взаємодії користувачів із системою, з тим, щоб найбільшою мірою утруднити чи виключити можливість реалізації загроз безпеці.

Організаційно-адміністративні заходи включають в себе :

розробку правил обробки інформації в ІС;

сукупність дій при проектуванні та обладнанні обчислювальних центрів та інших об'єктів ІС (облік впливу стихії, пожеж, охорона приміщень тощо);

сукупність дій при підборі й підготовці персоналу (перевірка нових співробітників, ознайомлення їх з порядком роботи з конфіденційною інформацією, з мірами відповідальності за порушення правил її обробки; створення умов, при яких персоналу було б невигідно допускати зловживання тощо);

організацію надійного пропускного режиму;

організацію обліку, зберігання, використання та знищення документів і носіїв з конфіденційною інформацією;

розподіл реквізитів розмежування доступу (паролів, повноважень тощо);

організацію прихованого контролю за роботою користувачів і персоналу ІС;

сукупність дій при проектуванні, розробці, ремонті та модифікації устаткування і програмного забезпечення (сертифікація використовуваних технічних і програмних засобів, суворе санкціонування, розгляд і затвердження

всіх змін, перевірка на задоволення вимогам захисту, документальна фіксація змін тощо).

До *фізичних мір* захисту відносяться різні механічні, електро- і електромеханічні пристрої або споруди, спеціально призначені для створення фізичних перешкод на можливих шляхах проникнення і доступу порушників (турнікети, колючий дріт, кодові замки, системи охоронно-пожежної сигналізації тощо).

До *апаратно-програмним заходам* захисту відносяться різні електронні пристрої та спеціальні програми, які реалізують самостійно або в комплексі з іншими засобами наступні способи захисту[5]:

ідентифікацію і аутентифікацію суб'єктів ІС;

розмежування доступу до ресурсів ІС;

контроль цілісності даних;

забезпечення конфіденційності даних;

аудит подій, що відбуваються в ІС;

резервування ресурсів і компонентів ІС.

Лекція 2.

Технічний захист інформації.

Інформаційна безпека є комплексом, в якому не можна виділити важливіші чи менш важливі складові. її не можна сприймати інакше, ніж комплекс.

Загрози інформаційній безпеці - чинник або сукупність чинників, що створюють небезпеку функціонуванню й розвитку інформаційного простору, інтересам особистості, суспільства, держави. Основним питанням початкового етапу впровадження системи безпеки є призначення відповідальних осіб за безпеку і розмежування сфер їх впливу. Системні програмісти та адміністратори відносять це завдання до компетенції загальної служби безпеки, тоді як остання вважає, що цим питанням мають займатися спеціалісти по комп'ютерах.

Вирішуючи питання розподілу відповідальності за безпеку комп'ютерної системи, слід ураховувати такі правила:

о ніхто, крім керівництва, не може прийняти основоположні рішення в галузі політики комп'ютерної безпеки;

о ніхто, крім спеціалістів, не зможе забезпечити правильне функціонування системи безпеки;

о ніяка зовнішня організація чи група спеціалістів життєво не зацікавлені в економічній ефективності заходів безпеки.

Організаційні заходи безпеки інформаційних систем прямо чи опосередковано пов'язані з адміністративним управлінням і належать до рішень і дій, які застосовує керівництво для створення таких умов експлуатації, які зведуть до мінімуму слабкість захисту. Адміністрація здійснює:

о заходи фізичного захисту комп'ютерних систем;

о регламентацію технологічних процесів;

о регламентацію роботи з конфіденційною інформацією;

о регламентацію процедур резервування;

о регламентацію внесення змін;

о регламентацію роботи персоналу й користувачів;

о підбір і підготовку персоналу;

о заходи контролю і спостереження.

До стратегічних рішень при створенні системи комп'ютерної безпеки потрібно віднести розроблення загальних вимог щодо класифікації даних, котрі зберігаються і опрацьовуються в системі.

Технічний захист інформації (ТЗІ) - діяльність, спрямована на забезпечення інженерно-технічними заходами конфіденційності, цілісності та доступності інформації.

Технічний захист секретної інформації спрямований на забезпечення інженерно-технічними заходами конфіденційності, цілісності та унеможливлення блокування інформації.

Консультативні послуги в галузі технічного захисту інформації - розроблення та надання рекомендацій щодо організації (створення) комплексу технічного захисту інформації об'єкта інформаційної діяльності або інформаційної системи на підставі матеріалів їх дослідження; надання методичної допомоги в розробленні нормативно-правових актів, нормативних документів системи ТЗІ, проектної і робочої документації при створенні засобів або комплексів технічного захисту інформації, проведення експертизи таких документів.

Технічний захист інформації є важливим чинником реалізації організаційно-правових та інженерно-технічних заходів з метою запобігання витоку інформації за рахунок несанкціонованого доступу до неї, несанкціонованим діям та впливам на інформацію, які призводять до її знищення, порушення цілісності або блокування, а також протидії технічним розвідкам.

Слід дотримуватися заходів захисту в усіх точках мережі, за будь-якої роботи суб'єктів з корпоративною інформацією.

Правову основу технічного захисту інформації в Україні становлять:

о Конституція України;

о закони України;

о міжнародні договори України;

о угоди, обов'язковість виконання яких введена Верховною Радою України;

о укази Президента України;

о постанови Кабінету Міністрів України;

о розпорядження адміністрації Державної служби спеціального зв'язку та захисту інформації України;

о інші нормативно-правові акти з питань технічного захисту інформації.

Правову основу створення і діяльності ПЗІ становлять:

о Закон України "Про державну таємницю";

о Закон України "Про захист інформації в автоматизованих системах";

о Положення про технічний захист інформації в Україні;

о Положення про забезпечення режиму секретності під час оброблення інформації, що становить державну таємницю, в автоматизованих системах;

о інші нормативно-правові акти з питань захисту інформації;

о державні і галузеві стандарти;

о розпорядчі та інші документи.

Підрозділ захисту інформації (ПЗІ) здійснює діяльність відповідно до "Плану захисту інформації", календарних, перспективних та інших планів робіт, затверджених керівництвом компанії. Проте виконання будь-яких завдань структурними підрозділами залежить від суб'єктів системи технічного захисту, якості їхньої підготовки, професіоналізму, матеріального забезпечення і чіткої взаємодії з іншими структурами компанії та органами контролю.

Під суб'єктом (рис. 6.18) у цьому разі розуміють користувача системи, процес, комп'ютер або програмне забезпечення для оброблення інформації. Кожен інформаційний ресурс (комп'ютер користувача, сервер організації або мережеве устаткування) має бути захищений від усіх можливих загроз.

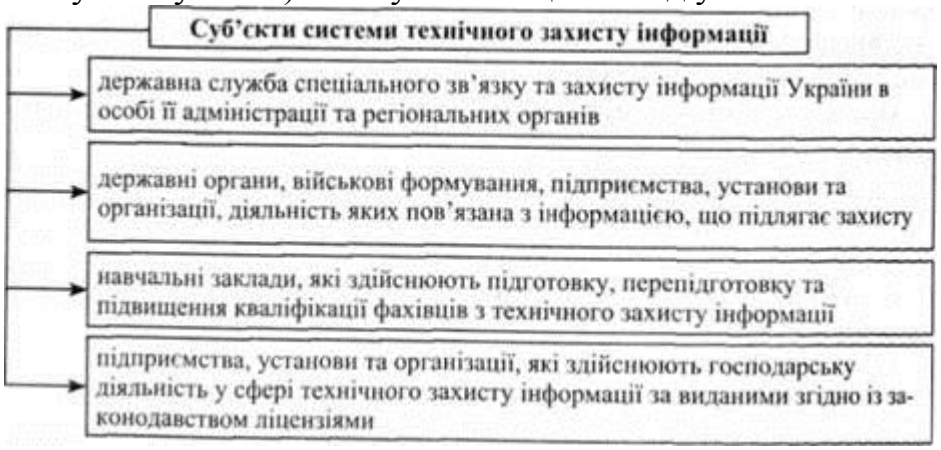


Рис. 6.18. **Суб'єкти системи технічного захисту інформації**

Державна політика у сфері технічного захисту інформації формується згідно із законодавством і реалізується Держспецзв'язком у взаємодії з іншими суб'єктами системи технічного захисту інформації.

Метою створення ПЗІ є організаційне забезпечення завдань керування комплексною системою захисту інформації (КСЗІ) на підприємстві та здійснення контролю за її функціонуванням. На ПЗІ покладається виконання робіт з:

о визначення вимог щодо захисту інформації в автоматизованій інформаційній системі підприємства (АІС);

о проектування;

- розроблення і модернізації КСЗІ;

о експлуатації;

о обслуговування;

о підтримки працездатності КСЗІ;

о контролю за станом захищеності інформації в комп'ютерних системах (КС).

Для проведення окремих заходів захисту інформації в КС, що пов'язані з напрямом діяльності інших підрозділів компанії, наказом керівництва визначають перелік, строки виконання робіт та виконавців - підрозділи або конкретних осіб. У своїй роботі ПЗІ взаємодіє з підрозділами компанії

(режимно-секретним відділом, службою безпеки, відділом ділової розвідки, службою охорони та ін.), а також з державними органами, установами та організаціями, що займаються питаннями захисту інформації.

У разі потреби до виконання робіт можуть бути залучені зовнішні організації, що мають ліцензії на відповідний вид діяльності у сфері захисту інформації.

У будь-якому каналі зв'язку виникають перешкоди, що призводять до спотворення інформації, яка надходить для опрацювання. Для зменшення вірогідності помилок вживають заходів щодо поліпшення технічних характеристик каналів, використання різних видів модуляції, розширення пропускної спроможності та ін. При цьому також потрібно вживати заходів щодо захисту інформації від помилок або несанкціонованого доступу.

Доступ - це надання можливості використовувати інформацію, що зберігається в ЕОМ (системі).

Будь-яка інформація в машині або системі потребує певного захисту, під яким розуміють сукупність методів управління доступом виконуваних у системі програм до інформації, що зберігається в ній.

Захисту підлягає будь-яка документована інформація, неправомірне поводження з якою може завдати збитку її власникові, користувачеві чи іншій особі.

Режим захисту інформації встановлюють щодо:

о відомостей, віднесених до державної таємниці уповноваженими органами на підставі чинного законодавства;

о конфіденційної документованої інформації власника інформаційних ресурсів або уповноваженою особою на законних підставах;

о персональних даних.

Завданнями підрозділу захисту інформації є:

1. Забезпечення безпеки інформації структурних підрозділів та персоналу компанії в процесі інформаційної діяльності та взаємодії між собою, а також у взаємовідносинах із зовнішніми вітчизняними та закордонними організаціями.

2. Дослідження технології опрацювання інформації з метою виявлення:

* можливих каналів витоку та інших загроз для безпеки інформації;

* формування моделі загроз; розроблення політики безпеки інформації;

* вивчення заходів щодо її реалізації.

3. Організація та координація робіт, пов'язаних із захистом інформації в компанії, необхідність захисту якої визначається чинним законодавством.

4. Підтримка необхідного рівня захищеності інформації, ресурсів і технологій.

5. Розроблення проектів нормативних і розпорядчих документів, чинних у межах організації, згідно з якими має бути забезпечений захист інформації в компанії.

6. Організація робіт зі створення і використання КСЗІ на всіх етапах життєвого циклу КС.

7. Участь в організації професійної підготовки і підвищенні кваліфікації персоналу та користувачів КС з питань захисту інформації.

8. Формування у персоналу і користувачів компанії розуміння необхідності виконання вимог нормативно-правових актів, нормативних і розпорядчих документів, що стосуються сфери захисту інформації.

9. Організація забезпечення виконання персоналом і користувачами вимог нормативно-правових актів, нормативних і розпорядчих документів із захисту інформації компанії.

10. Проведення контрольних перевірок виконання персоналом і користувачами вимог нормативно-правових актів, нормативних і розпорядчих документів із захисту інформації компанії.

11. Забезпечення визначених політикою безпеки властивостей інформації під час створення та експлуатації КС.

12. Своєчасне виявлення та знешкодження загроз для ресурсів КС, причин і умов порушення її функціонування та розвитку.

13. Створення механізму та умов оперативного реагування на загрози для безпеки інформації, інші прояви негативних тенденцій у функціонуванні КС.

14. Ефективне знешкодження загроз для ресурсів КС або запобігання їм шляхом проведення комплексу правових, морально-етичних, фізичних, організаційних, технічних та інших заходів гарантування безпеки.

15. Керування засобами захисту інформації, керування доступом користувачів до ресурсів КС, контроль за їхньою роботою з боку персоналу ПЗІ, оперативне сповіщення про спроби НСД до ресурсів КС підприємства.

16. Реєстрація, збирання, зберігання, опрацювання даних про всі події в системі, які стосуються безпеки інформації.

17. Створення умов для максимально можливого відшкодування та локалізації збитків, завданих несанкціонованими діями фізичних та юридичних осіб, впливом зовнішнього середовища та іншими чинниками.

18. Зменшення негативного впливу наслідків порушення безпеки на функціонування КС.

Основними загрозами безпеці інформації і нормального функціонування 1С є такі:

- просочування конфіденційної інформації;
- компрометація інформації;
- несанкціоноване використання інформаційних ресурсів;
- помилкове використання інформаційних ресурсів;
- несанкціонований обмін інформацією між абонентами;
- відмова від інформації;
- порушення інформаційного обслуговування;
- незаконне використання привілеїв.

Просочування конфіденційної інформації - це її безконтрольний вихід за межі ІС або через коло осіб, яким вона була довірена за видом служби або стала відома в процесі роботи. Цей витік може бути наслідком:

- розголошування конфіденційної інформації;
- витоку інформації різними, переважно технічними каналами;
- несанкціонованого доступу до конфіденційної інформації різними способами.

Розголошування інформації, що призвело до ознайомлення з нею осіб, не допущених до цих відомостей, можна кваліфікувати як умисні або необережні

дії посадових осіб і користувачів, яким ці відомості були довірені у зв'язку зі службовою потребою. Можливий безконтрольний витік конфіденційної інформації візуально-оптичним, акустичним, електромагнітним та іншими каналами.

Несанкціонований доступ - це протиправне навмисне оволодіння конфіденційною інформацією особою, яка не має права доступу до відомостей, що охороняються. Найпоширенішими напрямами несанкціонованого доступу до інформації є:

о перехоплення електронних випромінювань;

о примусове електромагнітне опромінювання (підсвічування) ліній зв'язку з метою отримання паразитної модуляції;

о застосування підслуховуючих пристроїв (жучків);

о дистанційне фотографування;

о перехоплення акустичних випромінювань і відновлення тексту принтера;

о зчитування залишкової інформації в пам'яті системи після виконання санкціонованих запитів;

о копіювання носіїв інформації з подоланням заходів захисту;

о маскування під зареєстрованого користувача;

о маскування під запити системи;

о використання програмних пасток;

о використання недоліків мов програмування і операційних систем;

о незаконне підключення до апаратури і ліній зв'язку спеціально розроблених апаратних засобів, що забезпечують доступ інформації;

о зловмисне виведення з ладу механізмів захисту;

о розшифровування спеціальними програмами зашифрованої інформації;

о інформаційні інфекції.

Перелічені напрями несанкціонованого доступу потребують значних технічних знань і відповідних апаратних або програмних розробок з боку зломлювача. Використовують, наприклад, технічні канали витоку - фізичні шляхи від джерела конфіденційної інформації до зловмисника, за допомогою яких можна отримати відомості, що охороняються. Причиною виникнення каналів витоку є конструктивна й технологічна недосконалість схематичних рішень або експлуатаційне спрацювання елементів. Все це дає змогу зломлювачу робити перетворювачі, що діють за певними фізичними принципами і мають властивий цим принципам канал передачі інформації - канал витоку.

Під час створення та експлуатації КСЗІ компанії підрозділ захисту інформації виконує такі функції:

1. Організація процесу керування КСЗІ.

2. Розслідування випадків порушення політики безпеки, небезпечних та непередбачених подій, здійснення аналізу причин, що призвели до них, супроводження банку даних таких подій.

3. Вжиття заходів у разі виявлення спроб НСД до ресурсів КС, порушення правил експлуатації засобів захисту інформації або інших дестабілізаційних факторів.

4. Забезпечення контролю цілісності засобів захисту інформації та швидке реагування на їх вихід із ладу або порушення режимів функціонування.

5. Організація керування доступом до ресурсів КС - розподіл між користувачами необхідних реквізитів захисту інформації:

о паролів;

о привілеїв;

о ключів та ін.

6. Супроводження й активізація бази даних захисту інформації:

о матриці доступу;

о класифікаційні мітки об'єктів;

о ідентифікатори користувачів тощо.

7. Спостереження (реєстрація і аудит подій в КС, моніторинг подій тощо) за функціонуванням КСЗІ та їх компонентів.

8. Підготовка пропозицій щодо удосконалення порядку забезпечення захисту інформації в КС, впровадження нових технологій захисту і модернізації КСЗІ.

9. Організація і проведення заходів з модернізації, тестування, оперативного відновлення функціонування КСЗІ після збоїв, відмов, аварій КС або КСЗІ.

10. Участь у роботах з модернізації КС:

о узгодженні пропозицій щодо введення до складу КС нових компонентів;

о нових функціональних завдань;

о режимів оброблення інформації, заміни засобів оброблення інформації тощо.

11. Забезпечення супроводження й активізації еталонних, архівних і резервних копій програмних компонентів КСЗІ, забезпечення їх зберігання і тестування.

12. Проведення аналітичного оцінювання поточного стану безпеки інформації в КС:

о прогнозування виникнення нових загроз та їх врахування в моделі загроз;

о визначення необхідності її коригування;

о аналіз відповідності технології оброблення інформації;

о аналіз реалізованої політики безпеки поточної моделі загроз та ін.

13. Доведення власникам інформації технічних можливостей захисту інформації в КС і типові правила для персоналу і користувачів КС.

14. Негайне втручання в процес роботи КС у разі виявлення атаки на КСЗІ, проведення у таких випадках робіт з викриття порушника.

15. Регулярне подання звітів керівництву компанії-власника (розпорядника) КС про виконання користувачами КС вимог захисту інформації.

16. Аналіз відомостей про технічні засоби захисту інформації нового покоління.

17. Обґрунтування пропозицій щодо придбання засобів для компанії.

18. Контроль за виконанням персоналом і користувачами КС вимог, норм, правил, інструкцій щодо захисту інформації відповідно до визначеної політики її безпеки.

19. Контроль забезпечення режиму секретності у разі оброблення в КС інформації, що становить державну таємницю.

20. Контроль забезпечення охорони і порядку зберігання документів (носіїв інформації), які містять відомості, що піддягають захисту.

21. Розроблення і реалізація спільно з РСВ компанії комплексних заходів безпеки інформації під час проведення заходів з науково-технічного, економічного, інформаційного співпраці з іноземними фірмами.

22. Розроблення і реалізація спільно з РСВ компанії комплексних заходів безпеки інформації під час проведення нарад, переговорів тощо, здійснення їх технічного та інформаційного забезпечення.

Більшість систем захисту в таких випадках використовують набори привілеїв, тобто для виконання певної функції потрібний певний привілей. Зазвичай користувачі мають мінімальний набір привілеїв, адміністратори - максимальний.

Набори привілеїв охороняються системою захисту. Несанкціоноване (незаконне) захоплення привілеїв можливе за наявності помилок у системі захисту, але здебільшого - в процесі керування системою захисту, зокрема у разі недбалого користування привілеями.

Чітке дотримання правил керування системою захисту, принципу мінімуму привілеїв дає змогу уникнути таких порушень.

Лекція 3.

Міжнародні стандарти у галузі інформаційної безпеки.

1. Стандарти і специфікації в галузі безпеки інформаційних систем.
2. «Помаранчева книга» як оцінний стандарт.
3. Класи безпеки інформаційних систем.
4. Технічна специфікація X.800 Стандарт ISO/IEC 15408. Розвиток стандартів з управління ризиками. Стандарт ISO/IEC TR 13335

ISO/IEC 17799

ISO/IEC 17799 2005 призначений для використання будь-якою організацією, котра планує встановити систему ефективного інформаційного захисту або покращувати існуючі методи інформаційного захисту.

Однак, це не свідчить, що всі рекомендації стандарту повинні бути обов’язково прийняті. Все залежить від конкретних місцевих інформаційних ризиків та вимог.

Стандарт складається з 13 розділів:

1. Загальна частина
2. Терміни та визначення
3. Політика безпеки
4. Організовані методи забезпечення інформаційної безпеки
5. Управління ресурсами
6. Користувачі інформаційної системи
7. Фізична безпека
8. Управління комунікаціями та процесами

9. Контроль доступу
10. Придбання та розробка інформаційних систем
11. Управління інцидентами інформаційної безпеки
12. Управління безперервністю ведення бізнесу
13. Відповідність вимогам

Кожен із розділів має таку структуру:

Мета – вказує, яка мета повинна бути досягнута

Управління – вказує, як цілі можуть бути досягнуті

Керівництво – вказує, як управління може бути реалізовано та **Додатки.**

Зокрема в термінах і визначеннях позиціонуються такі поняття, як:

інформаційна безпека (збереження конфіденційності, цілісності й доступності інформації), *конфіденційність* (забезпечення доступу до інформації тільки для авторизованих користувачів, що мають право на доступ до неї), *цілісність* (захист точності й повноти інформації й методів її обробки), *доступність* (забезпечення доступності інформації й пов'язаних з нею ресурсів авторизованим користувачам за необхідності) тощо.

Також визначається *політика безпеки*. Опис політики інформаційної безпеки – документ, що містить опис політики інформаційної безпеки, повинен бути схвалений керівництвом, опублікований й відповідно до необхідності розповсюджений серед всіх співробітників.

ISO/IES 27001

ISO/IEC 27001 – міжнародний стандарт по інформаційної безпеки розроблений спільно Міжнародною Організацією по Стандартизації (ISO) і Міжнародної електротехнічної комісією (IEC). Підготовлено до випуску підкомітетом SC27 Об'єднаного технічного комітету JTC 1.

Стандарт містить вимоги в області інформаційної безпеки для створення, розвитку і підтримки Системи менеджменту інформаційної безпеки.

Кращі світові практики в галузі управління інформаційною безпекою описані в міжнародному стандарті на системи менеджменту інформаційної безпеки ISO / IEC 27001 (ISO 27001). ISO 27001 встановлює вимоги до системи менеджменту інформаційної безпеки (СМІБ) для демонстрації здатності організації захищати свої інформаційні ресурси.

Поняття "захисту інформації" трактується міжнародним стандартом як забезпечення конфіденційності, цілісності та доступності інформації.

Основа стандарту ICO 27001 – система управління ризиками, пов'язаними з інформацією. Система управління ризиками дозволяє отримувати відповіді на наступні питання: – На якому напрямку інформаційної безпеки потрібно зосередити увагу? – Скільки часу і коштів можна витратити на дане технічне рішення для захисту інформації?

ISO/IES 15408

Стандарт ISO/IEC 15408 «Загальні критерії оцінки безпеки інформаційних технологій» (англ. Common Criteria for Information Technology Security Evaluation) описує інфраструктуру (Framework) в якій користувачі комп'ютерної системи можуть описати вимоги, розробники можуть заявити про властивості безпеки продуктів, а експерти з безпеки визначити, чи задовольняє

продукт заявам. Таким чином цей стандарт дозволяє бути впевненим, що процес опису, розробки та перевірки продукту був проведений в строгому порядку. Прообразом даного документа послужили «Критерії оцінки безпеки інформаційних технологій» (англ. Evaluation Criteria for IT Security, ECITS), робота над якими почалася в 1990 році.

Стандарт містить два основних види вимог безпеки: функціональні, що висуваються до функцій безпеки і реалізує їх механізмів, і вимоги довіри, які пред'являються до технології та процесу розробки та експлуатації.

Лекція 4.

Теоретико-числові базиси.

1. Унітарні функції та коди.
2. Функції Хаара та розрядно-позиційні коди.
3. Дискретно-фазові функції та коди Лібова-Крейга.

До основних дискретних теоретико-числових базисів належать унітарні функції та коди, функції Хаара та розрядно-позиційні коди, дискретно-фазові функції та коди Лібова-Крейга, функції Радемахера та двійкові коди, функції Грея та коди Грея, функції Уолша, функції Ґалуа та кодові системи Ґалуа. Вибір кодової системи, базису або системи функцій залежить від задачі, властивостей інформаційного потоку, умов застосування даних та інш.

Зокрема, базисами для виконання дискретних ортогональних перетворень і дискретного подання одновимірного інформаційного потоку зі скінченною енергією, визначеного в просторі $L_2[a,b]$ на часовому інтервалі $T=[a,b]$, є повні ортонормовані системи функцій. Вейвлет-аналіз здійснюється на основі ортогональних або базисів Ріса. Повними ортонормованими системами функцій у просторі $L_2[a,b]$ є тригонометрична система та дискретні експоненціальні функції – як базис перетворення Фур'є, системи Уолша, Хаара, функції пилкоподібного базису, Віленкіна-Крестенсона та інші, проте актуальною залишається задача визначення галузей, способів і методів їх ефективного застосування, формування та дослідження інших базисів. Визначення особливостей та ефективності застосування різних систем функцій для виконання дискретних перетворень і аналізу інформаційних потоків зумовлює необхідність дослідження властивостей цих систем, наведених у наступних викладках.

1. Унітарні функції та коди.

В якості вихідних у засобах перетворення форми інформації широкого застосування набули унітарні коди, розрядність бінарного подання слова яких відповідає повній шкалі квантування діапазону перетворення N. Здійснити перехід до ефективніших кодів із меншою розрядністю дозволяє аналітичне

подання унітарних кодів і встановлення функціональних залежностей з іншими кодами чи системами кодування.

Для подання унітарних кодів використовуються унітарні функції:

$$Uni(m,\theta,i) = sign(\sin(2^m \pi(\theta + i \cdot 2^{-n}))), \qquad (1.5)$$

де $m = 0, 1, \ldots, n+1$ – порядок набору системи функцій; $n = \log_2 N$; N – модуль цілочислових дискретних значень системи; $\theta = t/T$; $(0 \le \theta < 1)$ – нормований параметр часу; $T = 2\pi$; t – потокове значення часу; $0 \le t < 2\pi$; $i = 0,1,\ldots,2^{n-m+1} - 1$ – порядковий номер функції в наборі порядку m.

Набір нульового порядку $Uni(0,\theta,i)$ містить *2N* функцій (рис.1.1).

Властивості унітарних функцій:

1. Система з перших *N* унітарних функцій порядку *m* є лінійно незалежною, оскільки виконується достатня умова лінійної незалежності: ранг матриці *N* функцій дорівнює кількості функцій *N*. Наступні *N* функцій є лінійними комбінаціями *N* перших.

2. Унітарні функції не ортогональні, оскільки $\int_0^1 Uni(m,\theta,i)Uni(k,\theta,j)d\theta \neq 0$

.

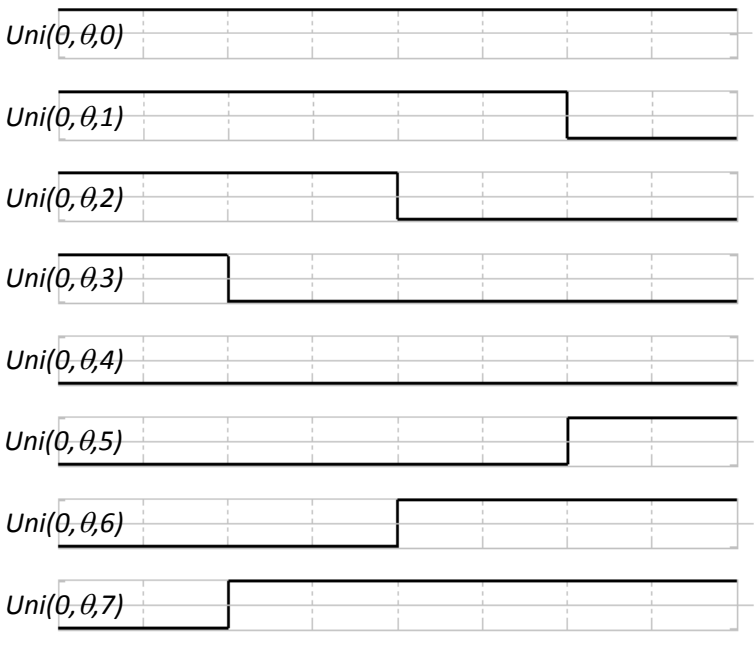


Рисунок 1.1 – Унітарні функції нульового порядку.

Неортогональність системи унітарних функцій зумовлює некомпактне пакування кодових елементів системи, що приводить до значної надлишковості інформаційних потоків. Внаслідок неортогональності та відсутності досліджень властивостей система не використовується як основа ТЧП.

Породжуючу кодову матрицю унітарного коду розміру $N \times N$ одержують при дискретизації з інтервалом $1/N$ за параметром часу перших $N=2^n$ із системи $2N$ унітарних функцій та здійсненні бінарної заміни значень функцій *1* на *0*, *–1* на *1* в точках $\theta_s = s/2^n$, $s=0,1,\ldots,2^n-1$, яка реалізується за допомогою операції:

$$u_i = (1 - Uni(0, \theta_s, 2^n - 1 - i))/2, \qquad (1.6)$$

де $u_0, u_1, \ldots, u_i, \ldots, u_{2^n-1}$ – значення розрядів унітарного коду θ_s, $i=0,1,\ldots,2^n-1$.

Для прикладу, при $n=3$ восьми функціям відповідають такі елементи кодової матриці:

$$\begin{array}{lcccccccc}
Uni(0,\theta,0) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Uni(0,\theta,1) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
Uni(0,\theta,2) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
Uni(0,\theta,3) & \rightarrow & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
Uni(0,\theta,4) & \rightarrow & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
Uni(0,\theta,5) & \rightarrow & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
Uni(0,\theta,6) & \rightarrow & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
Uni(0,\theta,7) & \rightarrow & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

Наведені властивості системи унітарних функцій дозволяють у наступних викладках визначити процедури перетворення до інших базисів. У ролі первинних при перетворенні форми інформації та при переході від *N*-розрядних унітарних до кодів із меншою розрядністю також використовуються розрядно-позиційні коди.

2. Функції Хаара та розрядно-позиційні коди.

Основою розрядно-позиційних кодів є система функцій Хаара. Функції Хаара $Har(n,\theta,j)$ (рис.1.2) визначаються за формулою:

$$Har(n,\theta,j)=\begin{cases}2^{\frac{n-1}{2}}\,sign(\sin 2^n\pi\theta), & \dfrac{j}{2^{n-1}}\le\theta<\dfrac{j+1}{2^{n-1}},\\ 0 \quad при\ інших\ \theta\in[0,1), & \end{cases} \tag{1.7}$$

де $n=0,1,\ldots,\log_2 N$; $j=0,1,\ldots,2^{n-1}-1$; ($j=0$ при $n=0$), $0\le\theta<1$.

При реалізації ТЧП використовуються наступні властивості системи Хаара.

1. Функції Хаара $\{Har(n,\theta,j)\}$ утворюють повну ортонормовану систему в просторі інтегровних із квадратом функцій $L_2[0,1)$, що дає можливість використовувати систему в якості базису для виконання ортогонального перетворення, яке є вейвлет-перетворенням.

2. На значній частині інтервалу визначення функції дорівнюють нулю, що дає можливість скоротити кількість арифметичних операцій при обчисленні перетворення. У результаті зменшується час обробки інформації.

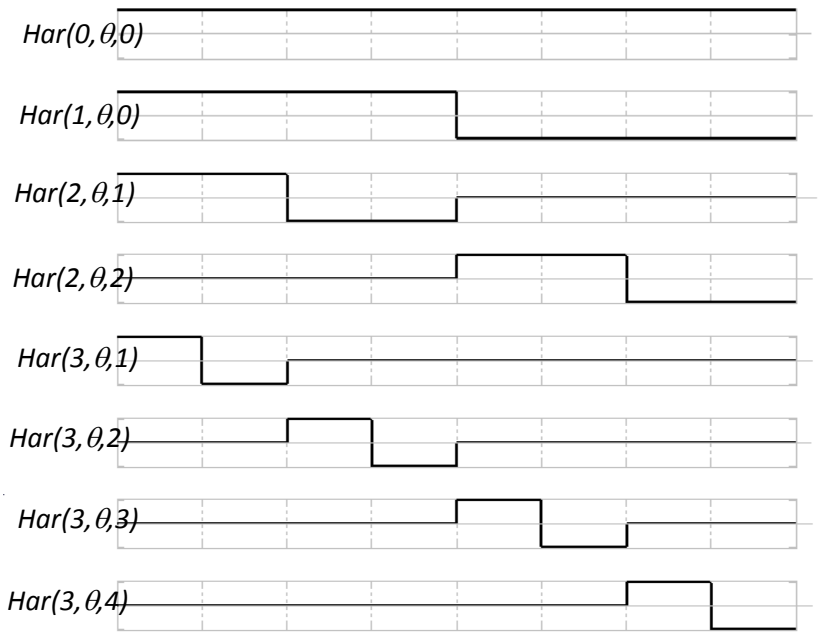


Рисунок 1.2 – Система функцій Хаара.

Ненормований базис Хаара $\{sign(\sin 2^n \pi\theta)\}$ (без нормуючого множника $2^{\frac{n-1}{2}}$), в якому функції набувають значень ± 1, 0, є основою розрядно-позиційних кодів. Розрядно-позиційні коди застосовуються в засобах перетворення форми інформації, в якості проміжних при аналогово-цифровому перетворенні, в давачах переміщень, для ініціювання комірок пам'яті тощо.

Для встановлення аналітичних співвідношень зв'язку систем функцій, які лежать в основі перетворення даних із розрядно-позиційного коду та в розрядно-позиційний код, використовуються розрядно-позиційні функції:

$$RP(i,\theta) = \begin{cases} -1, \dfrac{i}{2^n} \le \theta \le \dfrac{i+1}{2^n}, \\ 1 \ \textit{при інших} \ \theta \in [0,1], \end{cases} \qquad (1.8)$$

$n = 0,1,2,\ldots,\ i = 0,1,\ldots,2^n - 1.$

Дискретне подання 2^n розрядно-позиційних функцій та бінарна заміна значень функцій *1* на *0*, *–1* на *1*, яка подається за допомогою виразу:

$$p_i = (1 - RP(i,\theta_s))/2 = \frac{1}{2^{n/2}} Har(n+1,\theta_s,i-1), \qquad (1.9)$$

де p_i – значення розрядів розрядно-позиційного коду, породжує кодову матрицю:

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & 0 & 0 & \ldots & 1 & 0 \\ \hdashline 0 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 0 & 0 & \ldots & 0 & 0 \end{array}$$

За умови простої реалізації перетворення інформації унітарному та розрядно-позиційному кодам властивий недолік необхідності використання повної *N*-розрядної шини кодового подання даних для кодування *N* дискретних повідомлень. Це зумовлює необхідність переходу до ефективніших методів

кодування зі зменшеною розрядністю кодів до $n = \log_2 N$ в системах Радемахера та Грея.

В якості проміжних при переході до *n*-розрядних кодів використовуються коди Лібова-Крейга, які дозволяють у два рази зменшити розрядність коду та характеризуються властивістю абсолютного позиціонування.

3. Дискретно-фазові функції та коди Лібова-Крейга.

Залежність унітарних кодів з іншими встановлюється за допомогою системи дискретно-фазових функцій, що є основою кодів Лібова-Крейга.

Дискретно-фазові функції порядку m подаються згідно наступного аналітичного виразу:

$$Dyf(m,\theta,i) = sign(\sin(2^m \pi(\theta + i \cdot 2^{-n}))), \qquad (1.10)$$

де $i = 0, 1, \ldots, 2^{n-m+1}-1$ – порядковий номер функції в наборі порядку m.

Графіки дискретно-фазових функцій першого порядку наведені на рис.1.3

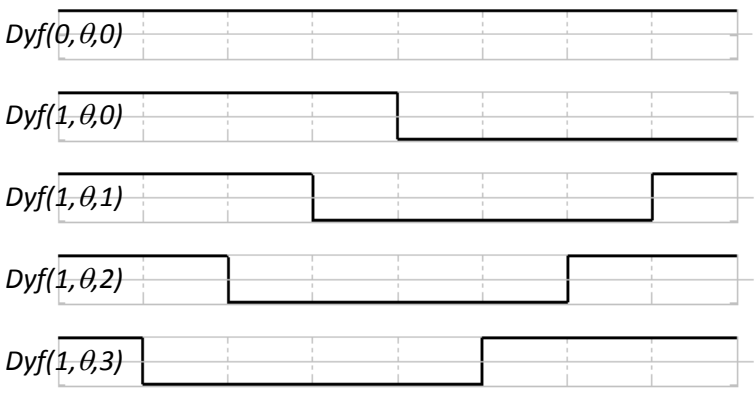
Властивості дискретно-фазових функцій:

1. Система з *N* дискретно-фазових функцій є лінійно залежною, оскільки частина функцій системи є лінійною комбінацією інших функцій системи:

$$Dyf(m,\theta, j + 2^{n-m}) = -Dyf(m,\theta, j),$$

де $j = 0,1,\ldots,2^{n-m} - 1.$

Внаслідок лінійної залежності перші *N* функцій не утворюють повної системи.



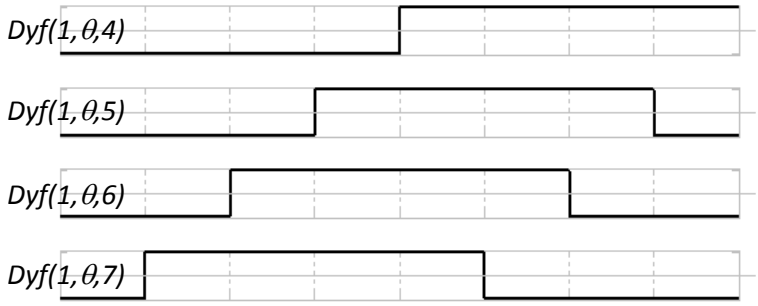


Рисунок 1.3 – Дискретно-фазові функції першого порядку.

2. Система є неортогональною, тому що $\int_0^1 Dyf(m,\theta,i)Dyf(m,\theta,j)d\theta \neq 0$.

Породжуючу кодову матрицю розміру $\frac{N}{2}\times N$ одержують за допомогою дискретизації перших $N=2^{n-1}$ дискретно-фазових функцій порядку m та здійснення бінарної заміни значень функцій 1 на 0, –1 на 1, яка реалізується за формулою:

$$d_j = (1 - Dyf(1,\theta_s,2^{n-1}-1-j))/2, \qquad (1.11)$$

де $d_0, d_1, ..., d_j, ..., d_{2^{n-1}-1}$ – значення розрядів коду Лібова-Крейга $\theta_s = \frac{s}{2^n}$, $s=0,1,...,2^n-1$.

Наприклад, при N=8 елементи матриці відповідатимуть таким функціям

$Dyf(1,\theta,0)$	→	0	0	0	0	1	1	1	1
$Dyf(1,\theta,1)$	→	0	0	0	1	1	1	1	0
$Dyf(1,\theta,2)$	→	0	0	1	1	1	1	0	0
$Dyf(1,\theta,3)$	→	0	1	1	1	1	0	0	0
s	→	0	1	2	3	4	5	6	7.

Внаслідок неповноти, система не використовується як основа для ортогональних ТЧП. Дискретно-фазові функції розглядаються як перехідні та як основа творення базисів і систем функцій Радемахера, Грея, кодування даних в яких здійснюється з розрядністю $n = \log_2 N$ порівняно з N для унітарних кодів. Із цією метою в складі системи дискретно-фазових функцій виокремлюють дві підсистеми функцій виду $sign(\sin(2^n \pi\theta))$ та $sign(\cos(2^n \pi\theta))$.

Лекція 5

Система функцій Радемахера, Грея, Уолша

1. Система функцій Радемахера та двійкові коди.
2. Система функцій Грея та коди Грея.
3. Система функцій Уолша.

1. Система функцій Радемахера та двійкові коди.

Екстракція *sin*-складових набору дискретно-фазових функцій утворює систему функцій Радемахера (рис. 1.4).

$$Rad(n,\theta) = Dyf(n,\theta,0) = sign(\sin(2^n \pi\theta)). \qquad (1.12)$$

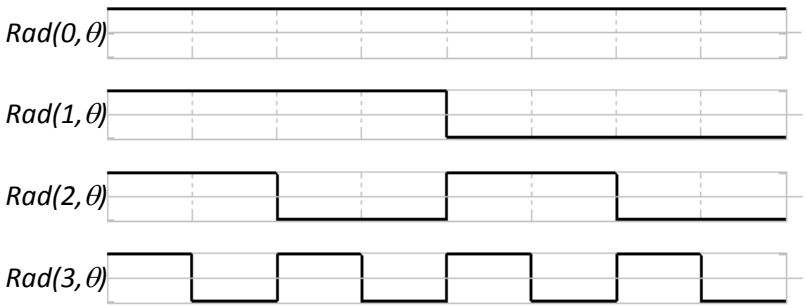


Рисунок 1.4 – Функції Радемахера.

Система Радемахера є основою двійкової системи числення.

Відповідність між значеннями функцій у точках $\theta_s = s/2^n$, $s=0,1,\ldots,2^n-1$ та їх поданням у двійковому коді $\theta_s = r_n r_{n-1} \ldots r_0$ встановлюється співвідношенням:

$$r_k = (1 - Rad(n-k, \theta_s))/2, \qquad (1.13)$$

де r_k – значення розрядів двійкового коду, $k = 0,1,\ldots,n$.

Наприклад, при $n=3$ чотирьом функціям відповідають такі елементи кодової матриці розміру 4×8

$$
\begin{array}{lcccccccc}
Rad(0,\theta) & \to & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Rad(1,\theta) & \to & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
Rad(2,\theta) & \to & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
Rad(3,\theta) & \to & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
s & \to & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7.
\end{array}
$$

Наступні властивості системи Радемахера:

- функції Радемахера ортонормовані на відрізку $[0,1)$, оскільки:

$$\int_0^1 Rad(n,\theta)Rad(k,\theta)d\theta = 0 \text{ та } \int_0^1 Rad(n,\theta)Rad(n,\theta)d\theta = 1.$$

- система функцій Радемахера утворює в просторі інтегровних із квадратом функцій $L_2[0,1)$ неповну систему ортонормованих функцій, оскільки для довільного n не виконується означення повноти системи:

$$\int_0^1 Rad(n,\theta)Rad(1,\theta)Rad(2,\theta)d\theta = 0,$$

тобто існує функція $Rad(1,\theta)Rad(2,\theta)$, яка тотожно не дорівнює нулю на інтервалі $[0,1)$ та ортогональна до всіх функцій системи.

Неповнота системи Радемахера обмежує її застосування для подання інформаційних потоків на основі ортогональних перетворень. Одночасно із широким застосуванням, творенням за допомогою системи Радемахера двійковим кодам властивий недолік, що полягає в неоднозначності формування відліків суміжних кодів при міжрозрядному позиціонуванні. Уникнути такої вади дозволяє перехід до кодів Грея

2. Система функцій Грея та коди Грея.

Екстракція *cos*-складових згідно кожного з порядків n набору дискретно-фазових функцій утворює систему функцій Грея. Система функцій Грея є підмножиною системи дискретно-фазових функцій:

$$Gry(0,\theta) = Dyf(0,\theta + 2^{-1},0);$$

$$Gry(m,\theta) = Dyf(m,\theta,2^{n-m-1}),\ m = 1,2,\ldots,n. \qquad (1.14)$$

Графіки функцій Грея наведено на рис.1.5.

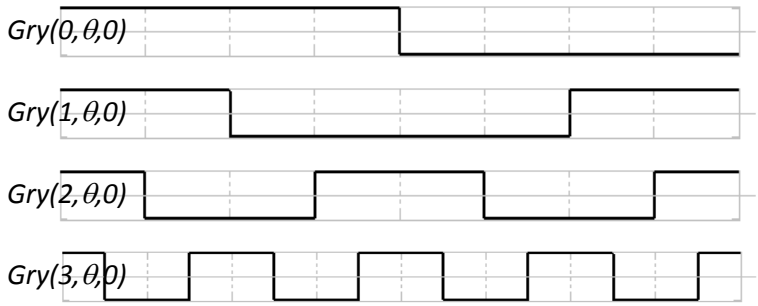


Рисунок 1.5 – Функції Грея.

Система функцій Грея є ортонормованою та неповною, що доводиться наступними викладками.

Відомо, що система Грея є складовою підсистемою функцій Уолша. Оскільки система функцій Уолша є ортонормованою, то і функції Грея, як її підсистема, є ортонормованими, тобто

$$\int_0^1 Gry(n,\theta)Gry(k,\theta)d\theta = 0,\ \int_0^1 Gry(k,\theta)Gry(k,\theta)d\theta = 1.$$

Функції Грея утворюють у $L_2[0,1)$ неповну систему, оскільки існують функції, які тотожно не дорівнюють нулю та ортогональні до всіх функцій системи, зокрема, для довільного n:

$$\int_0^1 Gry(n,\theta)Rad(2,\theta)d\theta = 0.$$

Неповнота системи Грея обмежує її застосування для розкладання інформаційних потоків та реалізації ТЧП.

Розширення функціональних можливостей при реалізації системних функцій перетворення форми та обробки інформації забезпечує перехід до базису Уолша. Аналітичні залежності процедури переходу базуються на наступній властивості скінченних добутків функцій системи Грея.

Якщо $(k_1,\ldots,k_m)$ і $(l_1,\ldots,l_r)$ дві різні скінченні послідовності, то:

$$\int_0^1 [Gry(k_1,\theta)..Gry(k_m,\theta)][Gry(l_1,\theta)..Gry(l_r,\theta)]d\theta = 0. \qquad (1.15)$$

Для доведення властивості необхідно перепозначити множники в підінтегральному виразі $Gry(j_1, \theta)$, $Gry(j_2, \theta)$,…, $Gry(j_p, \theta)$ $(j_1<j_2<\ldots<j_p)$. Добуток пар функцій $Gry(j_k, \theta)Gry(j_k, \theta)=1$. Добуток інших множників $Gry(j_1, \theta)\ Gry(j_2, \theta)\ldots\ Gry(j_{p-1}, \theta)$ є частково-сталою функцією, кожний з інтервалів сталості якої можна поділити на парне число рівних підінтервалів, на яких $Gry(j_p,\theta)$ набуває почергово значення $+1$ і -1 або -1 і $+1$. Із врахуванням чого:

$$\int_0^1 [Gry(j_1,\theta)..Gry(j_p,\theta)]d\theta = const\int_I Gry(j_p,\theta)d\theta = 0,$$

а тому буде рівним нулю значення інтегралу на інтервалі $[0;1)$. Тобто два різні добутки функцій системи є ортогональними, що і треба довести.

Система функцій Грея є основою кодів Грея. Відповідність між значеннями функцій у точках $\theta_s = s/2^n$, $s=0,1,\ldots,2^n-1$ та їх поданням у коді Грея $\theta_s = h_n h_{n-1}\ldots h_0$ встановлюється співвідношенням:

$$h_k = (1 - Gry(n-k-1,\theta_s))/2, \qquad (1.16)$$

де $k = 0,1,\ldots,n-1$.

Наприклад, чотирьом функціям відповідають елементи кодової матриці розміру 4×8

$$\begin{array}{cccccccccc}
Gry(0,\theta) & \to & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
Gry(1,\theta) & \to & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
Gry(2,\theta) & \to & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
Gry(3,\theta) & \to & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
s & \to & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7.
\end{array}$$

Кодування Грея дозволяє зменшити похибки формування відліків суміжних кодів унаслідок зміни тільки одного розряду порівняно із застосуванням двійкових кодів. Однак, неповнота систем функцій Радемахера та Грея звужує галузі їх ефективного застосування. Розширення функціональних можливостей при реалізації системних функцій перетворення форми та обробки інформації забезпечує базис Уолша.

3. Система функцій Уолша.

Властивості систем Радемахера, Грея та відповідних кодів визначають процедуру переходу в базис Уолша $Wal(i,\theta)$, $i = 0,1,\ldots,2^n - 1$, впорядкований за Уолшем, із системи Радемахера $Rad(n,\theta)$. Функції Уолша $Wal(i,\theta)$ (рис.1.6) визначаються, як добуток функцій Радемахера:

$$Wal(i,\theta) = Rad(1,\theta)^{b_0}\, Rad(2,\theta)^{b_1} \ldots Rad(n,\theta)^{b_{n-1}} = \prod_{k=0}^{n-1} (Rad(k+1,\theta))^{b_k}, \quad (1.17)$$

де $i = b_{n-1}b_{n-2}\ldots b_1 b_0$ – подання в коді Грея порядкового номера функції $Wal(i,\theta)$.

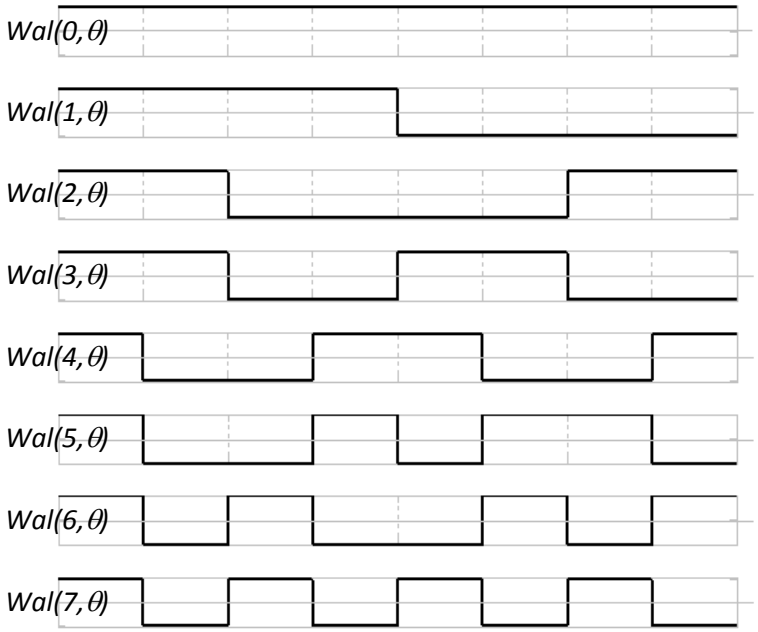


Рисунок 1.6 – Система функцій Уолша.

Система функцій Уолша є ортонормованою, повною та мультиплікативною.

Функції Уолша ортонормовані на інтервалі $0 \le \theta \le 1$

$$\int_0^1 Wal(k,\theta)Wal(i,\theta)d\theta = \begin{cases} 1 \ \ при \ \ k = i, \\ 0 \ \ при \ \ k \ne i. \end{cases}$$

Функції Уолша утворюють мультиплікативну систему

$$Wal(k,\theta)Wal(i,\theta) = Wal(k \oplus i,\theta).$$

Системи Радемахера та Грея є підсистемами функцій Уолша

$$Wal\,(2^n\!-\!1,\theta)= Rad(n,\theta),$$

$$Wal\,(2^n,\theta)= Gry(n,\theta).$$

Відомі схеми генераторів функцій Уолша базуються на методі формування функцій Уолша із системи Радемахера, але використання в ньому двійкового коду зумовлює виникнення помилки неоднозначності. На основі доведеної властивості (1.11) ортогональності добутків функцій Грея розроблено альтернативний метод формування функцій Уолша із системи функцій Грея:

$$Wal(i,\theta) = Gry(0,\theta)^{a_0}\, Gry(1,\theta)^{a_1} \ldots Gry(n\text{-}1,\theta)^{a_{n-1}} = \prod_{k=0}^{n-1} (Gry(k,\theta))^{a_k}\ , \quad (1.18)$$

де $i = a_{n-1}a_{n-2}\ldots a_1a_0$ – подання в двійковому коді порядкового номера функції Уолша $Wal(i,\theta)$.

У відомих генераторах і перетворювачах функції Уолша формуються згідно (1.17) на основі двійкового коду наступним способом. Функції Радемахера відповідають значенням розрядів двійкового коду:

$$Rad(m,\theta_s) = \overline{r_{n-m}} - r_{n-m} = \begin{cases} 1, & якщо\ r_{n-m} = 0, \\ -1, & якщо\ r_{n-m} = 1, \end{cases} \quad (1.19)$$

де r_m – значення m-го розряду двійкового коду аргумента θ_s, $m=0,1,...,n$.

При підстановці (1.20) у вираз (1.18) функції Уолша визначаються на основі двійкового коду аргументу θ_s:

$$\begin{gathered} Wal(i,\theta_s) = (\overline{r_{n-1}} - r_{n-1})^{b_0} (\overline{r_{n-2}} - r_{n-2})^{b_{n1}} \ldots (\overline{r_0} - r_0)^{b_{n-1}} = \\ = (\overline{b_0 r_{n-1} \oplus b_1 r_{n-2} \oplus \ldots \oplus b_{n-1} r_0}) - (b_0 r_{n-1} \oplus b_1 r_{n-2} \oplus \ldots \oplus b_{n-1} r_0) =, \\ = \overline{\varphi_i}(\theta_s) - \varphi_i(\theta_s) \end{gathered} \quad (1.20)$$

де $\varphi_i(\theta_s) = b_0 r_{n-1} \oplus b_1 r_{n-2} \oplus \ldots \oplus b_{n-1} r_0$.

Із використанням наведеної методики можна визначити спосіб формування функцій Уолша на основі аргумента, поданого в коді Грея.

Функції Грея $Gry(k,\theta_s)$ відповідають значенням *n* розрядів коду Грея θ_s згідно залежності:

$$Gry(k,\theta_s) = \overline{h_{n\text{-}k\text{-}1}} \text{ - } h_{n\text{-}k\text{-}1} = \begin{cases} 1, & якщо\ h_{n-k-1} = 0, \\ -1, & якщо\ h_{n-k-1} = 1, \end{cases} \qquad (1.21)$$

де h_k – значення *k*-го розряду коду Грея аргумента θ_s; $k = 0,1,\ldots,n-1$.

Функції Уолша визначаються при підстановці функцій Грея з (1.21) у (1.18)

$$Wal(i,\theta_s) = (\overline{h_{n-1}} - h_{n-1})^{a_0} (\overline{h_{n-2}} - h_{n-2})^{a_1} \ldots (\overline{h_0} - h_0)^{a_{n-1}},$$

де $\theta_s = h_{n-1}\ldots h_1 h_0$ – код Грея, $i = a_{n-1}a_{n-2}..a_0$ – подання у двійковому коді числа *i*.

Перетворення добутку в правій частині рівності з використанням основних тотожних співвідношень булевої алгебри дозволяє визначити функції Уолша:

$$Wal(i,\theta_s) = \overline{(a_0 h_{n-1} \oplus a_1 h_{n-2} \oplus \ldots \oplus a_{n-1} h_0)} - (a_0 h_{n-1} \oplus a_1 h_{n-2} \oplus \ldots \oplus a_{n-1} h_0), \qquad (1.22)$$

$$Wal(i,\theta_s) = \overline{\varphi_i}(\theta_s) - \varphi_i(\theta_s), \qquad (1.23)$$

де $\varphi_i(\theta_s) = a_0 h_{n-1} \oplus a_1 h_{n-2} \oplus \ldots \oplus a_{n-1} h_0$.

Перевагою перетворювача на основі (1.22) є використання кодів Грея, які дозволяють зменшити помилки в засобах перетворення та обробки.

Таким чином, повна система Уолша, яка утворюється із систем Радемахера та Грея, є базисом для виконання ортогональних перетворень і формування функцій Ґалуа.

Лекція 6.

Система функцій Ґалуа та кодові системи Ґалуа та Крестенсона.

1. Система функцій Ґалуа та кодові системи Ґалуа
2. Система функцій Крестенсона

Перехід до різних упорядкувань функцій у системі Ґалуа здійснюється з базису Уолша з упорядкуванням функцій за рекурсивним законом. За *n*-розрядними фрагментами рекурсивної послідовності, яка утворюється відповідно до породжуючого вектора поля Ґалуа $GF(2^n)$, згідно відображення через систему функцій Радемахера формуються номери функцій Уолша та Ґалуа в системі.

Наприклад, у полі $GF(2^3)$ існують породжуючі вектори 1011 та 1101. Для даного поля $GF(2^3)$ рекурсивні послідовності $v_0, v_1, v_2, v_3, v_4, \ldots$ формуються з початкового вектора $(v_0 v_1 v_2) = (111)$ за правилами:

1) $1101 \rightarrow v_{i+3} = v_i \oplus v_{i+1}$: $v_0, v_1, v_2, v_0 \oplus v_1, v_1 \oplus v_2, v_0 \oplus v_1 \oplus v_2, v_0 \oplus v_2, v_0, v_1, v_2, \ldots$;

2) $1011 \rightarrow v_{i+3} = v_i \oplus v_{i+2}$:

$v_0, v_1, v_2, v_0 \oplus v_2, v_0 \oplus v_1 \oplus v_2, v_0 \oplus v_1, v_1 \oplus v_2, v_0, v_1, v_2, \ldots$.

Для початкового вектора $(v_0 v_1 v_2) = (111)$ утворена на основі породжуючого вектора 1101 рекурсивна послідовність {0 0 0 1 0 1 1 1}, визначає наступне рекурсивне впорядкування номерів функцій Уолша в системі {0 1 2 5 3 7 6 4}.

0	0	0	1	0	1	1	1	0	0	
0	0	0							→	0
	0	0	1						→	1
		0	1	0					→	2
			1	0	1				→	5
				0	1	1			→	3
					1	1	1		→	7
						1	1	0	→	6
							1	0	0 →	4

Утворена на основі породжуючого вектора 1011 рекурсивна послідовність {0 0 0 1 1 1 0 1} визначає інше рекурсивне впорядкування номерів функцій Уолша: {0 1 3 7 6 5 2 4}.

Із рекурсивно впорядкованої системи Уолша відповідно впорядковані перші *n* функцій Ґалуа формуються згідно співвідношення:

$$Gal(n,\theta,i) = Wal(Ent(2^n\theta), \frac{2^{i+1}-1}{2^n}), \qquad (1.24)$$

де $i = 0,1,\ldots,2^n - 1$, *Ent*– функція виділення цілої частини.

Проведені дослідження встановили можливість формування функцій Ґалуа із систем Радемахера та Грея. Згідно співвідношень (1.17) та (1.18) перші *n* функцій Ґалуа в системі подаються у вигляді добутку функцій Радемахера та Грея:

$$Gal(n,\theta,i) = \prod_{k=0}^{n-1}(Rad(k+1, \frac{2^{i+1}-1}{2^n}))^{h_k} = \prod_{k=0}^{n-1}(Gry(k+1, \frac{2^{i+1}-1}{2^n}))^{r_k}, \qquad (1.25)$$

де $h_{n-1}h_{n-2}\ldots h_0$ – запис у коді Грея числа *q*, двійковий код якого є *n*-розрядним фрагментом $v_i v_{i+1} v_{i+2}\ldots v_{i+n-1}$ рекурсивної послідовності $v_0, v_1, v_2, \ldots$; $r_{n-1}r_{n-2}\ldots r_0$ – двійковий код, який є *n*-розрядним фрагментом $v_i v_{i+1} v_{i+2}\ldots v_{i+n-1}$ рекурсивної послідовності $v_0, v_1, v_2, \ldots$.

Повний набір 2^n функцій рекурсивної системи Ґалуа $Gal(n,\theta,i)$ отримують із перших *n* функцій системи процедурою рекурсивного зсуву на $\Delta\theta = \frac{1}{2^n}$ згідно другої діагоналі кожної наступної функції відносно попередньої:

$$Gal(n,\theta,i+1) = Gal(n,\theta+\Delta\theta,i). \qquad (1.26)$$

Впорядкування функцій Ґалуа в наборі відповідає синтезованому за породжуючим вектором упорядкуванню функцій Уолша.

Процедура переходу від дискретних значень функцій Уолша до дискретних значень функцій Ґалуа подається матричною операцією:

$$\|Gal\| = \|W\| \cdot \|R\|,$$

де $\|Gal\|$ – матриця розміру $N \times n$ системи Ґалуа; $\|W\|$ – матриця розміру $N \times N$ рекурсивно впорядкованих функцій Уолша; $\|R\|$ – матриця розміру $N \times n$ відображеної вагової мережі Радемахера.

Для прикладу, матрична операція переходу від функцій Уолша до функцій Ґалуа та матриця розміру 8×8 дискретних значень функцій Ґалуа в полі $GF(2^3)$ з породжуючим вектором 1101 згідно процедури рекурсивного розширення подаються відповідно:

$$\begin{Vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{Vmatrix} \cdot \begin{Vmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{Vmatrix} = \begin{Vmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & 1 \end{Vmatrix} \longrightarrow \begin{Vmatrix} 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 \end{Vmatrix}.$$

Графіки функцій Ґалуа $Gal(n, \theta, i)$ з породжуючим вектором 1101 при $n=3$ наведено на рис.1.7.

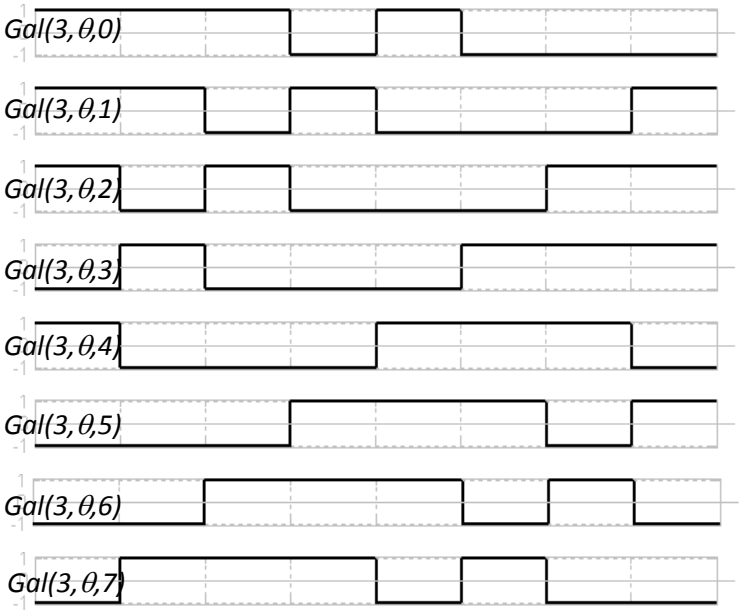


Рисунок 1.7 – Функції Ґалуа $Gal(3,\theta,i)$ з породжуючим вектором 1101.

При виконанні перетворень використовуються наступні властивості системи функцій Ґалуа.

1. У системі 2^n функцій n-го порядку кожна підсистема із n функцій $\{Gal(n,\theta,i),\ Gal(n,\theta,i+1), Gal(n,\theta,i+2),\ldots, Gal(n,\theta,i+n-1)\}$ ортогональна.

2. Симетрія індекса та аргумента. Елементи матриці системи Ґалуа симетричні відносно головної діагоналі:

$$Gal(n,\theta_s,i) = Gal(n,\frac{i}{2^n},s),$$

де $i,s \in \{0,1,\ldots.2^n-1\}$.

Таким чином, матриці системи Ґалуа є ганкелевими антициклічними, оскільки при $i+j=k+l$ $G_{ij}=G_{kl}$. Якщо рекурсивний зсув у (1.26) здійснюється згідно головної діагоналі та у формулі (1.26) $\Delta\theta = -\frac{1}{2^n}$, то матриці є тоепліцевими циклічними (циркулянтними), оскільки $i+j=k+l$ $G_{ij}=G_{kl}$.

3. Міри довжин інтервалів, на яких $Gal(n,\theta_s,i)=1$ і $Gal(n,\theta_s,i)=-1$ однакові, отже:

$$\int_0^1 Gal(n,\theta,i)d\theta = \sum_{s=0}^{2^n-1} Gal(n,\theta_s,i) = 0, \qquad (1.27)$$

тобто множина функцій Ґалуа задовольняє необхідну умову для вейвлет-функцій.

При дискретизації за параметром часу перших n функцій Ґалуа та здійсненні бінарної заміни значень функцій 1 на 0, –1 на 1, згідно виразу:

$$g_k(\theta_s) = (1 - Gal(n-k-1,\theta_s))/2, \qquad (1.28)$$

одержують матрицю кодових елементів Ґалуа розміру $n \times N$, $k = 0,1,\ldots,n-1$.

Наприклад, при N=8 рядки матриці кодових елементів Ґалуа з породжуючим вектором 1101 відповідатимуть таким функціям:

$$\begin{array}{cccccccccc} Gal(3,\theta,0) & \rightarrow & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ Gal(3,\theta,1) & \rightarrow & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ Gal(3,\theta,2) & \rightarrow & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ s & \rightarrow & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

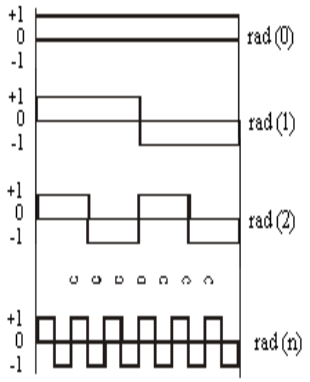
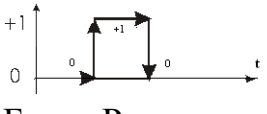
Елементи матриці кодових елементів Ґалуа з породжуючим вектором 1011 відповідатимуть наступним функціям:

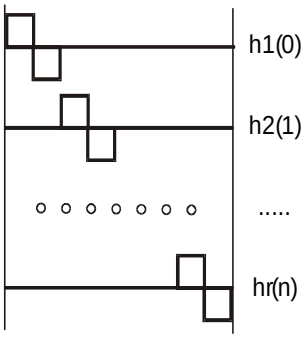
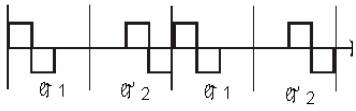
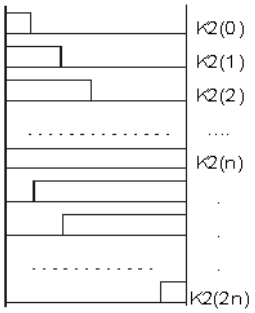
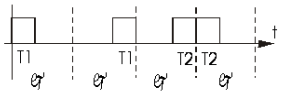
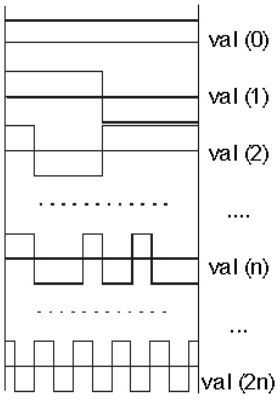
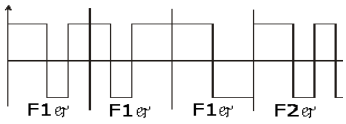
$$\begin{array}{cccccccccc} Gal(3,\theta,0) & \rightarrow & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ Gal(3,\theta,1) & \rightarrow & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ Gal(3,\theta,2) & \rightarrow & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ s & \rightarrow & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

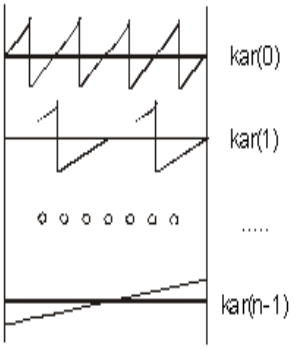
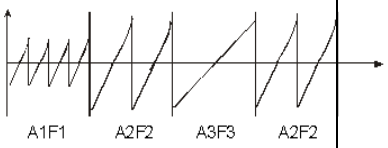
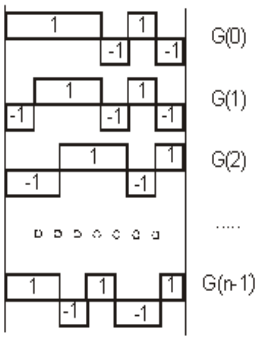
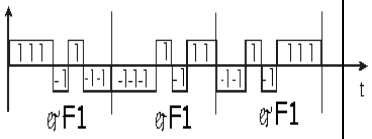
При дискретизації системи N функцій Ґалуа $\{Gal(n,\theta,i)\}$, $i = 0,1,\ldots.2^n - 1$ та перетворенні значень функцій згідно (5.30) отримують повну матрицю кодових елементів Ґалуа розміру $N \times N$, впорядкованих із поелементним рекурсивним зсувом згідно другої діагоналі матриці Ґалуа. Номер s повідомлення однозначно визначається n-координатним вектором $n = \log_2 N$.

Представлення систем ортогональних функцій різних ТЧБ подані у табл.1.1.

Таблиця 1.1 – Представлення теоретико-числових базисів

Базис	Представлення базису	Базисна функція та об’єм матриці V	Модуляція сигналу та спектр
Рademахера		Rad(n,θ)=sign[2^n π·θ] V=N·$\log_2$ N	Базисні функції Радемахера є основою для модуляції Прямокутних сигналів  Базис Радемахера

			породжує двійкову систему числення і двійкові коди
Хаара		$Har(n,\theta,i) =$ $= sign\left[\sin(i2^n\pi,\theta)\right]$ V=N^2	Використовується при фазовій модуляції сигналу.  В даному базисі використовуються розрядно-позиційні коди.
Крейга		Crg(n,θ)=sign[sin((2^n-1)·π·θ)] $V = \frac{N^2}{2}$	Породжує тривалісні і фазовіметоди модуляції сигналу. 
Уолша		Had (h,x) = $\prod_{i=1}^{\kappa}[ri\,(x)]hi$	Породжує частотно-фазові методи модуляції сигналу.  Базис Уолша має найбільш широкий спектр сигналу

Крестенсона		$N_i = res\sum_{i=1}^{n}(B_i \cdot b_i) \bmod P$ $V = \sum_{i=1}^{m} \log_2(P_i)$	Даний базис породжує амплітудно-частотні методи модуляції.  Базис представлений трикутними функціями. Спектр сигналів такого базису є експоненціальний.
Галуа		$N_j=f(C_{j-n-1},\ldots C_{j-1},C_j)$, $C_j = \sum_{j=0}^{n=1} C_{j-1} \cdot A \cdot (\bmod 2)$ V=N	Базис Галуа породжує коди поля Галуа і систему числення Галуа.  Модуляція в базисі як фазова, так і частотна.

З метою оцінки ефективності кодування даних на основі різних ТЧБ доцільно провести аналіз кодових матриць, які породжують різні системи числення.

При цьому важливою характеристикою кожного базису є об’єм його кодової матриці M_j та число активних елементів m_j (рис.1.7), що визначає характеристики надлишковості представлення інформації на основі аналітичної оцінки:

$$V_i = n_i \cdot N_i,$$

де n_j – розрядність числа; N_i – число незалежних кодових значень.

$$M_{Uni} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix} \quad а)$$

$$M_{har} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix} \quad б)$$

$$M_{Gr} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 0 \\ 1 & 1 & 1 & \dots & 0 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \end{vmatrix} \quad в)$$

$$M_{Rad} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{vmatrix} \quad г)$$

$$M_{LibCr} = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & & 0 & 1 \end{vmatrix} \quad д)$$

$$M_{Cres} = \begin{vmatrix} P_1 & P_2 & \dots & P_n \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ 2 & 2 & \dots & 2 \\ 0 & 3 & \dots & 3 \\ 1 & 4 & \dots & 4 \\ 2 & 0 & \dots & 5 \\ 0 & 1 & \dots & 6 \\ \dots & \dots & \dots & \dots \\ a_1 & a_2 & \dots & a_n \end{vmatrix} \quad е)$$

$$M_{Gal} = \begin{vmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{vmatrix} \quad є)$$

Рисунок 1.7 – Кодові матриці дискретних базисів:
а) унітарного; б) Хаара; в) Грея; г) Радемахера; д) Крейга; е) Крестенсона; є) Галуа.

Кожен з названих базисів характеризується визначеним об’ємом кодової матриці для представлення даних. При цьому найбільш надлишковим базисом є унітарний, в якого кодова матриця $V = N^2$, а число активних кодових елементів $n = N^2/2$, де N – діапазон кодування даних. Аналогічну надлишковість забезпечує базис Хаара, в два рази меншу надлишковість забезпечує базис Крейга, тобто $V = N^2/4$, а $n = N^2/8$. Максимально широке

застосування для кодування даних в сучасних КС отримали базиси Радемахера та Крестенсона, в яких $V = N \log_2 N$. ДаФні базиси відповідно породжують двійкову систему числення та систему числення залишкових класів.

Базис Уолша максимально широко використовується в сучасних телекомунікаційних КС. Даний базис породжує систему ортогональних шумоподібних сигналів, які використовуються в сотових системах мобільного зв'язку.

Найменшу надлишковість кодування даних забезпечує базис Галуа, кодова матриця якого $V = N$, а $n = N/2$.

Згідно викладеного, характеристики ТЧБ кодування даних, як системного об'єкта, подані в табл.1.2.

Таблиця 1.2 – Характеристики потоків даних

Формувачі вхідних та вихідних інформаційних сигналів даних	Характеристики інформаційних потоків даних
Унітарний базис	$V = N^2;\ n = N^2/2$
Базис Хаара	$V = N^2,\ n = N$
Базис Крейга	$V = N^2/4,\ n = N^2/8$
Базис Радемахера	$V = N \cdot \log_2 N,\ n = \frac{N}{2}\log_2 N$
Базис Крестенсона	$V = N \cdot \log_2 N$
Базис Уолша	$V = N^2, n = N^2/2$
Базис Галуа	$V = N,\ n = N/2$

Світовий досвід створення процесорів для комп'ютерних систем за останні 50 років, поряд з застосуванням теоретико-числового базису (ТЧБ) Радемахера, який породжує двійкову систему числення, демонструє тенденцію все ширшого застосування інших ТЧБ, в тому числі: унітарного, Хаара, Крестенсона та Галуа. Реалізація спеціалізованих, сигнальних, комутаційних та проблемно-орієнтованих процесорів цифрової обробки даних часто

виконується на базі сумісного використання комбінацій названих ТЧБ, наприклад Радемахера-Хаара, Крестонсона-Галуа та ін [56].

Перспективним напрямком розвитку теорії та технологій побудови спеціалізованих програмно-апаратних комп'ютерних засобів є реалізація супершвидкодіючих мультибазисних RCG-процесорів на основі базисів Радемахера, Крестенсона і Галуа [56]. Відомі успішні спроби розвитку теорії та техніки побудови матричних процесорів на основі двовимірних базисів Радемахера та Галуа, а також конвеєрних спецпроцесорів у базисі Галуа [54].

Спостережувані тенденції розвитку теорії методології та техніки процесорів комп'ютерних систем обумовлені теоретичним та ідейним насиченням можливостей застосування базису Радемахера для побудови арифметико-логічних компонентів процесорів, до яких ставляться все жорсткіші вимоги щодо швидкодії, покращення регулярності структури та розширення функціональних можливостей.

У зв'язку з цим існує проблема глибокого дослідження характеристик «нерадемахівських» ТЧБ та граничних можливостей їх застосування для реалізації компонентів як спеціалізованих, так і універсальних процесорів. При цьому перспективним, крім найбільш сьогодні масового одновимірного (векторного) представлення чисел та виконання арифметико-логічних операцій у базисі Радемахера перспективним є застосування двовимірних систем числення, вертикальної інформаційної технології у базисі Галуа та різних форм багатовимірного представлення чисел у вигляді залишків різних форм системи залишкових класів базису Крестенсона [2, 50].

Лекція 7.

Цілочисельна, нормалізована та досконала форми системи залишкових класів.

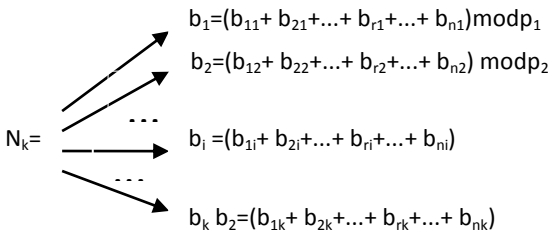
Система числення залишкових класів (СЗК) базису Крестенсона, розроблена Акушским І.Я. та Юдіцким Д.І. [59], особливо її цілочисельна форма, широко використовувалась починаючи з 70-х років минулого століття для побудови швидкодіючих спеціалізованих процесорів систем повітряної оборони колишнього СРСР.

Нормалізована форма СЗК, запропонована науковою школою проф. Николайчука Я.М., широко використана в телекомунікаційних процесорах інформаційних систем нафтогазової промисловості [57].

В роботі Николайчука Я.М., Возної Н.Я., Волинського О.І. [55] запропоновано чотири аналітичні моделі прямих та зворотніх перетворень залишкових класів (табл.1)

Таблиця 1 – Аналітичні моделі прямих та зворотніх перетворень залишкових класів

№ п\ п	Пряме перетворення форми СЗК	Зворотнє перетворення форми СЗК
1.	Цілочисельна форма СЗК	
	$N_k = (b_1 b_2 .. b_i .. b_k)_{(p_1 p_2 ... p_i ... p_k)}$ $N_k = b_i (\mathrm{mod}\, p_i)$, $N_k = a_i p_i + b_i$, $P = \prod_{i=1}^{k} p_i$; $0 \le N_k \le P$.	$b_i = res N_k (\mathrm{mod}\, p_i)$ $\quad N_k = res \sum_{i=1}^{k} b_i \cdot B_i (\mathrm{mod}\, P)$, $B_i = \frac{P}{p_i} \cdot m_i \equiv 1 (\mathrm{mod}\, P_i)$.
2.	Нормалізована форма СЗК	
	$\frac{N_k}{P} = res \sum_{i=1}^{k} \frac{b_i \cdot B_i (\mathrm{mod}\, P)}{P}$, $[N_k]_0 = res \sum_{i=1}^{k} b_i \cdot \frac{B_i}{P} (\mathrm{mod} 1)$, $0 \le [N_k]_0 \le P - 1$; $\frac{B_i}{P} = \frac{1}{p_i}$,	$[N_k]_0 = res \sum_{i=1}^{k} b_i \cdot \frac{m_i}{p_i} (\mathrm{mod} 1)$ $[N_k]_0 = res \sum_{i=1}^{k} [b_i]_0 \cdot m_i (\mathrm{mod} 1)$, $[b_i]_0 = \frac{b_i}{p_i}$, $0 \le [b_i]_0 \le 1$.

	$\delta_p \le \frac{1}{P}, \frac{1}{p_i} = 0.\overbrace{gggg}^{n_i}\overbrace{gggg}^{\delta_p},$	$N_k = \text{int}[N_k]_0 \cdot P,$
3.	Досконала форма СЗК	
	$[N_k]_0 = res\sum_{i=1}^{k}[b_i]_0 (\text{mod}1).$	$b_i = \text{int}\, res[N_k]_0 (\text{mod}1)\cdot P_i$
4.	Розмежована форма СЗК	
	$N_k = N_{1k} + N_{2k} + ... + N_{ik} + ... + N_{nk},$	

де N_k – число у позиційній системі числення (у базисі Радемахера); ($b_1 b_2 .. b_i .. b_k$) – представлення числа у СЗК; $_{(p_1 p_2 ... p_i ... p_k)}$ - набір взаємно простих модулів СЗК; b_i - найменший невід'ємний залишок; P – діапазон кодування чисел в СЗК; a_i – ранг; K – число модулів СЗК; B_i – базисні числа СЗК; res – символ операції знаходження найменшого невід'ємного залишку; int – символ операції виділення цілої частини; mod - символ операції по модулю; m_i – ранговий коефіцієнт СЗК; δр – дробова частина в нормалізованій формі СЗК; $[N_k]_0$, $[b_i]_0$ – відповідно число та залишок в нормалізованій формі базису Радемахера.

В даний час виконуються активні дослідження двовимірних (матричних) форм систем числення базисів Радемахера та Галуа [57].

В роботах Возної Н.Я., Николайчука Я.М. наведено алгоритми формування на низових рівнях КС структуризованих даних в цілочисельній формі базису Крестенсена, на низових рівнях КС в нормалізованій досконалій формі, в базизі Галуа у вигляді асинхронного алгоритму та синхронного структуризованого потоку даних рекурентних кодів Галуа з доповненням біт-орієнтованих інформаційних потоків ТЕД (табл. 2).

Таблиця 2 – Алгоритми формування на низових рівнях КС структуризованих даних в різних ТЧБ.

№ п/п	Алгоритми формування структуризованого потоку даних	Аналітичне представлення алгоритмів формування структуризованого потоку даних
1.	на низових рівнях КС структуризованих даних в цілочисельній формі базису Крестенсена	$\left.\begin{array}{l} x_1(t)\to x_{i1}\to p_1\to b_1 \\ x_2(t)\to x_{i2}\to p_2\to b_2 \\ \ldots \\ x_j(t)\to x_{ij}\to p_j\to b_j \\ \ldots \\ x_m(t)\to x_{im}\to p_{k-1}\to b_{k-1} \\ D_i\to p_k\to b_k \end{array}\right\} N_k = res\sum_{j=1}^{k} b_j B_j (\mathrm{mod} P);$ $B_j = \frac{P}{p_j} m_j = 1(\mathrm{mod} P_j)$
2.	на низових рівнях КС в нормалізованій досконалій формі	$\left.\begin{array}{l} x_1(t)\to x_{i1}\to p_1\to [b_1]_0 \\ x_2(t)\to x_{i2}\to p_2\to [b_2]_0 \\ \ldots \\ x_j(t)\to x_{ij}\to p_j\to [b_j]_0 \\ \ldots \\ x_m(t)\to x_{im}\to p_{k-1}\to [b_{k-1}]_0 \\ D_i\to p_k\to [b_k]_0 \end{array}\right\} [N_k]_0 = res\sum_{i=1}^{k}[b_i]_0 (\mathrm{mod}1)$ $b_i = \mathrm{int}\, res[N_k]_0 (\mathrm{mod}1)\cdot P$
3.	в базизі Галуа у вигляді асинхронного алгоритму	$\left.\begin{array}{l} x_1(t)\to G_{1i}\to G_{1,i-1}\ldots G_{1,i-n}\ldots G_{1,i-(n+k)} \\ x_2(t)\to G_{2i}\to G_{2,i-1}\ldots G_{2,i-n}\ldots G_{2,i-(n+k)} \\ \ldots\ldots \\ x_m(t)\to G_{mi}\to G_{m,i-1}\ldots G_{m,i-n}\ldots G_{1,i-(n+k)} \\ D_{oi}\to G_{oi} \end{array}\right\} \Rightarrow G_1, G_2, \ldots G_j \ldots G_m, G_{oi}$
4.	синхронного структуризованого потоку даних рекурентних кодів Галуа з доповненням біт-орієнтованих інформаційних потоків ТЕД	$\left.\begin{array}{l} x_1(t)\to (G_{1,i-1}\ldots G_{1,i-n}\ldots G_{1,i-(n+k)})\to G_{x1} \\ x_2(t)\to (G_{2,i-1}\ldots G_{2,i-n}\ldots G_{2,i-(n+k)})\to G_{x2} \\ \ldots\ldots \\ x_m(t)\to (G_{m,i-1}\ldots G_{m,i-n}\ldots G_{1,i-(n+k)})\to G_{xm} \\ D_{oi}\to G_{oi}\to G_{0m+1} \end{array}\right\} \Rightarrow G_{x1}, G_{x2}, \ldots G_{xj} \ldots G_{km}, G_{o,m+1}$

де $x_j(t)$ - аналогові дані телеметрії; x_{ij} - цифрові дані телеметрії; D_i – ТЕД;

p_1, p_2… p_k – система взаємно простих модулів; b_1, b_2….. b_j…. b_k – набір найменших невід'ємних залишків; Bj – система ортогональних базисів СЗК;

$[N_k]_0$ - аналітичний вираз з табл. 8.1.

Аналіз наукових тенденцій розвитку теорії та перспективних інформаційних технологій покращення ефективності опрацювання інформаційних потоків в комп'ютерних мережах, проведений на основі новітніх публікацій потребує поглибленого дослідження теоретичних засад базисів Крестенсона та Галуа. Слід зауважити, що найбільш фундаментально досліджено цілочисельну форму в системі залишкових класів, яка утворюється на основі прямого перетворення ТЧБ Крестенсона.

Тому є доцільним дослідити інші форми систем залишкових класів які можуть бути використані для реалізації високопродуктивних алгоритмів опрацювання і захисту інформаційних потоків, а також виконати порівняльний аналіз названих ТЧБ з базисом Радемахера, який породжує двійкову систему числення на основі відповідних критеріїв.

Відомо, що двійкова система числення, яка використовується в сучасних комп'ютерних системах, має певні недоліки – наявність міжрозрядних зв'язків та велику розрядність [46]. Тому актуальним є розвиток і застосування непозиційних систем числення, в яких відсутні вказані недоліки. Прикладом може бути система залишкових класів (СЗК), або, як її ще називають, представлення чисел у базисі Крестенсона [29], [47]. Хоча вона не набула значного поширення у зв'язку з необхідністю визначення умов переповнення, складністю та громіздкістю зворотнього перетворення чисел у десяткову систему числення, а також складнощами реалізації операцій ділення та порівняння, але СЗК можна ефективно використовувати у мультибазисних процесорах, спеціалізованих часових машинах для виконання операцій додавання, віднімання та множення, наприклад, у задачах лінійної алгебри (матрично–векторні операції) тощо. Необхідно відмітити, що ця система особливо ефективна при обчисленнях з великими числами [37], [48].

Фундаментальною основою СЗК є теорія чисел [51], [52], зокрема, властивості китайської теореми про залишки. Будь–яке ціле додатнє число N у десятковій системі числення представляється в СЗК у вигляді набору найменших додатніх залишків від ділення цього числа на фіксовані цілі додатні

попарно взаємно простими числа $p_1, p_2, \dots, p_n$ ($N_{10}=(b_1, b_2, \dots, b_n)_{p1, p2, \dots, pn}$, де $b_i=N \bmod p_i$), які називаються модулями (n – кількість модулів). При цьому повинна виконуватись умова $0 \le N \le P-1$, де $P=\prod_{i=1}^{n} p_i$.

На відміну від позиційних систем числення, де величина визначеного розряду суми, різниці або множення залежить не тільки від значень відповідних, але і від попередніх розрядів доданків або множників, в СЗК додавання, віднімання та множення цілих чисел виконується окремо по кожному модулю і переноси між розрядами відсутні. Отже, такі операції в СЗК є модульними [54].

Нехай два десяткові числа A і B, записані в СЗК за вибраними модулями: $A_{10}=(a_1, a_2, \dots, a_i, \dots, a_n)_{p1, p2, \dots, pi, \dots, pn}$, $B_{10}=(b_1, b_2, \dots, b_i, \dots, b_n)_{p1, p2, \dots, pi, \dots, pn}$. Тоді:

$A_{10} \pm B_{10}=C_{10}=(c_1, c_2, \dots, c_i, \dots, c_n)_{p1, p2, \dots, pi, \dots, pn}$;

$A_{10} \times B_{10}=D_{10}=(d_1, d_2, \dots, d_i, \dots, d_n)_{p1, p2, \dots, pi, \dots, pn}$,

де $c_i= a_i \pm b_i$, $d_i= a_i \times b_i$.

Останні рівності справедливі лише в тому випадку, коли результат операції не виходить за межі інтервала $\prod_{i=1}^{n} p_i - 1$.

Зворотнє перетворення із базису Крестенсона у десяткову систему числення є досить громіздким і грунтується на використанні китайської теореми про залишки [51]:

$$N=\left(\sum_{i=1}^{n} b_i B_i\right) mod\, P, \qquad (8.1)$$

де $B_i=M_i m_i$, $M_i=\frac{P}{p_i}$, m_i шукається з виразу $(M_i m_i) mod\, p_i = 1$, при цьому повинна виконуватись умова $\left(\sum_{i=1}^{n} B_i\right) mod\, P = 1$.

Слід зазначити, що при переведенні чисел із СЗК у десяткову систему числення значну часову складність становить пошук коефіцієнтів

$m_i = M_i^{-1} \, mod \, p_i$. У роботі [29] було запропоновано досконалу форму СЗК (ДФ СЗК), у якій підбір модулів такий, що m_i=1, тобто

$$M_i \, mod \, p_i = 1. \tag{8.2}$$

Крім того, було підібрано декілька наборів для чотирьох та п’яти модулів ДФ СЗК. Подальший розвиток ДФ СЗК отримала у роботах [63, 64], у яких було встановлено правила побудови наборів з будь–якої кількості модулів ДФ СЗК для будь–якого діапазону десяткових чисел. Шукані модулі повинні отримуватися з такої умови:

$$\begin{cases} p_1 = 2 \\ p_i = p_1 p_2 \ldots p_{i-1} + 1, \; 1 < i < n. \\ p_n = p_1 p_2 \ldots p_{n-1} - 1. \end{cases} \tag{3}$$

Слід зазначити, що запропонована система не вичерпує всіх можливих наборів для базису Крестенсона при заданих n. Наприклад, при n=5 набір модулів, отриманий за допомогою системи (3), буде P_{51} = 2, 3, 7, 43, 1805. Однак відомі також набори P_{52} = 2, 3, 7, 83, 85 та P_{53} = 2, 3, 11, 17, 59. При n=6 набір модулів, отриманий з (7.3), буде таким: P_{61} = 2, 3, 7, 43, 1807, 3263441. Усі можливі набори модулів для ДФ СЗК базису Крестенсона при n=6, відповідні їм діапазони десяткових чисел та розрядність у двійковій системі наведені у таблиці 3. Як видно з таблиці 3, набір модулів, отриманий за допомогою системи (3), найоптимальніший, оскільки в цьому випадку величина P є максимальна, що дозволяє розглядати найбільший діапазон десяткових чисел. При цьому досягається зменшення розрядності вдвічі.

Крім того, у цих роботах запропонована напівдосконала форма СЗК (m_i=±1), яку зручно використовувати у випадку обмеженої кількості модулів та необхідності розгляду великих чисел. Порівняно з ДФ СЗК, це збільшує часову

складність, але вона менша, ніж при пошуку оберненого елемента $m_i = M_i^{-1} \, mod \, p_i$.

Таблиця 8.3–Можливі набори модулів при N=6 для ДФ СЗК та відповідні їм діапазони десяткових чисел (в дужках – розрядність у двійковій системі)

№	p_1, p_2	p_3	p_4	p_5	p_6	P
1	2, 3 (2)	7 (3)	43 (6)	1807 (11)	3263441 (22)	$1{,}0650050423922\times10^{13}$ (44)
2				1811 (11)	654133 (20)	$2{,}139450562578\times10^{12}$ (41)
3				1819 (11)	252701 (18)	$8{,}30151592914\times10^{11}$ (41)
4				1825 (11)	173471 (18)	$5{,}7175174245\times10^{11}$ (40)
5				1871 (11)	51985 (16)	$1{,}7565866661\times10^{11}$ (38)
6				1901 (11)	36139 (16)	$1{,}24072631634\times10^{11}$ (37)
7				1945 (11)	25271 (15)	$8{,}876868357\times10^{10}$ (37)
8				2053 (12)	15011 (14)	$5{,}5656554898\times10^{10}$ (36)
9				2167 (12)	10841 (14)	$4{,}2427359282\times10^{10}$ (36)
10				2501 (12)	6499 (13)	$2{,}9354722194\times10^{10}$ (35)
11				3041 (12)	4447 (13)	$2{,}4423128562\times10^{10}$ (35)
12				3611 (12)	3613 (12)	$2{,}3562056658\times10^{10}$ (35)
13			47 (6)	395 (9)	779729 (20)	$6{,}0797809317\times10^{10}$ (36)
14				481 (9)	2203 (12)	$2{,}091735282\times10^{9}$ (31)
15			53 (6)	271 (9)	799 (10)	$4{,}81993554\times10^{8}$ (29)
16			71 (7)	103 (7)	61429 (16)	$1{,}8867671634\times10^{10}$ (35)
17		11 (4)	23 (5)	31 (5)	47057 (16)	$2{,}214408306\times10^{9}$ (32)

Напівдосконала форма дозволяє побудувати систему з двох модулів, що неможливо у ДФ СЗК. Для цього необхідно вибрати два будь–які послідовні

числа p_1 та $p_2=p_1+1$, які завжди будуть взаємно простими, оскільки для них виконується умова:

$$\begin{cases}(p_1+1) mod\ p_1 = 1 \\ p_1\, mod(p_1+1) = -1. \end{cases} \quad (4)$$

Загальна система для визначення набору з будь–якої кількості модулів напівдосконалої форми СЗК має вигляд:

$$\begin{cases} p_2 = p_1 + 1 \\ p_i = p_1 p_2 \ldots p_{i-1} \pm 1, \end{cases} \quad (5)$$

де $i = 3, \ldots, n$.

Перспективними модифікаціями СЗК, які на даний час глибоко досліджуються науковою школою професора Я.М.Николайчука, є нормалізована та розмежована форми СЗК, описані в [65].

Лекція 8.

Найбільший спільний дільник з використання базису Крестенсона. Алгоритм Евкліда в розмежованій системі числення

1. Найбільший спільний дільник. Алгоритм Евкліда.

Знаходження найбільшого спільного дільника (НСД) є важливою фундаментальною задачею теорії чисел, успішне вирішення якої дозволяє вдосконалити алгоритми широкого класу прикладних задач, особливо задач захисту інформаційних потоків в комп'ютерних системах з використанням асиметричної криптографії (алгоритмів RSA, Рабіна, Ель–Гамаля, електронного цифрового підпису, дослідження порядку еліптичної кривої за допомогою алгоритму Шуфа). Це зумовлено необхідністю використання, як правило, взаємно простих чисел, НСД яких дорівнює 1.

В зв'язку з цим актуальною проблемою досліджень є розробка теоретичних основ пошуку НСД з використанням теоретико-числових базисів Радемахера та Крестенсона, застосування яких дозволяє зменшити часову складність.

Найбільш розповсюдженим методом знаходження НСД є один з найдавніших математичних алгоритмів – алгоритм Евкліда, згідно якого для знаходження НСД двох чисел необхідно декілька разів від більшого числа відняти менше, поки різниця не стане меншою від'ємника. Тоді цю ж саму процедуру потрібно виконати з від'ємником та різницею. Процес віднімання буде тривати до тих пір, поки від'ємник та різниця не стануть однакові. Оскільки числа, над якими виконуються операції, на кожному кроці зменшуються, то такий процес не може тривати нескінченно, а закінчиться через деяке число кроків.

Сучасний математичний запис алгоритму Евкліда має такий вигляд [6]: для будь–якого $a>b=r_0$, де a і b –цілі числа, виконується система рівнянь

$$
\begin{aligned}
& a = r_0 \cdot q_1 + r_1,\ 0 \le r_1 < r_0; \\
& r_0 = r_1 \cdot q_2 + r_2,\ 0 \le r_2 < r_1; \\
& \ldots\ldots\ldots\ldots\ldots\ldots \\
& r_{n-2} = r_{n-1} \cdot q_n + r_n,\ 0 \le r_n < r_{n-1}; \\
& r_{n-1} = r_n \cdot q_{n+1} + 0.
\end{aligned} \qquad (1)
$$

НСД (a, b) дорівнюватиме r_n, тобто останньому ненульовому члену послідовності r_i. Оскільки $r_0>r_1>r_2>\ldots>r_n>0$, то даний процес закінчиться щонайменше через b кроків, тобто потрібно виконати b ділень з остачею. Однак дана оцінка є незадовільною, оскільки для знаходження НСД двох чисел (або перевірки їх на простоту), які використовуються в сучасних асиметричних криптосистемах, затрачається дуже великий обсяг часу. В [7] показано, що для знаходження НСД за допомогою алгоритму Евкліда потрібно виконати не більше, ніж 5k операцій ділення з остачею, де k – кількість цифр в десятковому записі числа a. В [8] показано, що кількість кроків не перевищує $2 \cdot log_2 b + 1$. Інша оцінка випливає з теореми Ламе [9], згідно якої кількість кроків алгоритму Евкліда не перевищує $\left[log_{\Phi}\left(\sqrt{5}a\right)\right] - 2$, де $[\alpha]$–верхнє ціле α; $\Phi = \left(1+\sqrt{5}\right)/2$ – більший корінь характеристичного рівняння послідовності Фіббоначчі.

Незважаючи на вказані оцінки та простоту програмної реалізації алгоритму Евкліда, його часова складність залишається великою, оскільки операція ділення є досить трудомісткою. Розрахунки показують, що пошук НСД з використанням алгоритму Евкліда в базисі Радемахера характеризується часовою складністю щонайменше $17{,}5 \cdot n(n+1)^2$, де n – розрядність числа a в двійковому коді.

Іншим недоліком алгоритму Евкліда є послідовне виконання операції ділення одна за одною, тобто неможливість розпаралелення його роботи.

Перспективним напрямком вдосконалення досліджуваних алгоритмів є розробка теорії їх реалізації на основі розмежованої системи числення залишкових класів [11] – [12].

Для зменьшення часової складності потрібно розробляти нові високопродуктивні алгоритми знаходження НСД за допомогою розмежованої системи числення та базису Крестенсона.

2. Алгоритм Евкліда в розмежованій системі числення

Нехай потрібно знайти НСД чисел a і b:

$$a=a_{n-1}2^{n-1}+ \dots +a_i2^i+ \dots +a_12+a_0 \quad (2)$$

$$b=b_{n-1}2^{n-1}+ \dots +b_i2^i+ \dots +b_12+b_0, \quad (3)$$

причому $a>b=r_0$; a,b=0,1.

Виходячи з (9.1), $r_1=a \bmod b= (a_{n-1}2^{n-1}+ \dots +a_i2^i+ \dots +a_12+a_0) \bmod b= \left(\sum_{i=0}^{n-1}\left(a_i 2^i \bmod b\right)\right) \bmod b=\left(\sum_{i=0}^{n-1}\left(a_i r_{1i}\right)\right) \bmod b$, де $r_{1i}=2^i \bmod b$. Це означає, що шуканий залишок дорівнюватиме сумі тих степенів двійки, для яких відповідно a_i=1. Слід зазначити також, що два послідовні значення r_{1i} та $r_{1\ i+1}$ пов'язані рекурентним співвідношенням $r_{1\ i+1}=(2\cdot r_{1i}) \bmod b$. Для знаходження залишку за модулем b не обов'язково виконувати ділення з остачею, а можна обмежитися відніманням: якщо $r_{1\ i+1}<b$, то воно залишається незмінним, в іншому випадку $r_{1\ i+1}= r_{1\ i+1}-b$. Найпростіше реалізувати описаний крок алгоритму Евкліда в розмежованій системі числення за допомогою таблиці 1.

Таблиця 1 – Таблиця знаходження залишку a mod b.

a_{n-1}	a_{n-2}	…	a_i	…	a_2	a_1	a_0
r_{1n-1}	r_{1n-2}	…	$r_{1\ i}$	…	r_{12}	r_{11}	r_{10}

Згідно таблиці 1, r_1 шукається як сума r_{1i} за модулем b, над якими у верхньому рядку розміщено 1, тобто $r_1=\left(\sum_{i=0}^{n-1} r_{1i}\right) \bmod b$ при умові a_i=1.

Аналогічно будуємо таблицю 2.

Таблиця 2 – Таблиця знаходження залишку b mod r_1.

$b_{n-1}=r_{0\ n-1}$	$b_{n-2}=r_{0\ n-2}$	…	$b_i=r_{0\ i}$	…	$b_2=r_{0\ 2}$	$b_1==r_{0\ 1}$	$b_0=r_{0\ 0}$
r_{2n-1}	r_{2n-2}	…	$r_{2\ i}$	…	r_{22}	r_{21}	r_{20}

Відповідно $r_2 = \left(\sum_{i=0}^{n-1} r_{2i} \right) mod\, r_1$ при умові b_i=1.

Узагальнюючи отримані результати, запишемо вираз для знаходження будь–якого залишку:

$$r_j = \left(\sum_{i=1}^{n-1} r_{j-2i} r_{ji} \right) mod\, r_{j-1},$$

де $r_{j-2\,i}$=0,1; $r_{ji}=2^i \bmod r_{i-1}$.

Відмітимо, що кількість кроків стандартного алгоритму Евкліда та алгоритму Евкліда в розмежованій системі числення однакові. Однак часова складність виконання кожного кроку істотно зменшується і становить $\log_2 n/2$. Загальна часова складність алгоритму Евкліда в розмежованій системі числення оцінюється виразом $O(17{,}5n(log_2\, n/2))$. Крім того, він виключає можливість розпаралелення.

Запропонований алгоритм знаходження НСД грунтується на пошуку залишків від ділення чисел a і b (a > b) в розмежованій системі числення на всі прості числа до $\sqrt{b}$.

Спільним дільником чисел a та b буде модуль p_j^k, який шукається з умови:

$$\left(\sum_{i=0}^{n-1} a_{ij} \right) mod\; p_j^k = \left(\sum_{i=0}^{n-1} b_{ij} \right) mod\; p_j^k = 0, \qquad (4)$$

де $a_{ij} = a_i \cdot 2^i\, mod\, p_j^k$, $b_{ij} = b_i \cdot 2^i\, mod\, p_j^k$, p_j – просте число, менше $\sqrt{b}$, k =1,2,3, ... – степінь p_j.

Слід зазначити, що при k=1 і виконанні (4) перевіряється та ж умова при k=2 і т.д. Таким чином враховується спільний дільник, який є степенем простого числа. Шуканий найбільший спільний дільник знаходиться як добуток отриманих за допомогою (4) спільних дільників.

Запропонований алгоритм характеризується логарифмічною часовою складністю $O(m \cdot log_2 n)$, де $m = \int_2^{\sqrt{b}} \frac{dt}{lnt}$ – кількість простих чисел в діапазоні від 2 до $\sqrt{b}$. Крім того, він дозволяє розпаралелити виконання всіх операцій по кожному модулю та виконати факторизацію чисел a та b.

Запропонований вище алгоритм можна суттєво удосконалити шляхом скорочення кількості модулів (тобто кількості кроків), за якими потрібно шукати залишки. Нехай маємо систему модулів, яка складається з простих чисел p_1, p_2, p_3, ..., менших $\sqrt{b}$, і для деякого p_j^k виконується умова (4). Наступний крок полягає у послідовній перевірці умови (4) для модуля $\left(p_j^k p_{j+1}^{k_1}\right)$, k_1=1,2,3,…; i=1,2,….

Таким чином, при послідовному домноженні модулів отримуємо, що НСД(a, b)= $\prod_{j=1}^{s} p_j^k$, для яких виконуєься умова (4).

В порівнянні з попереднім, даний алгоритм використовує меншу кількість кроків, однак він не піддається розпаралеленню і не вирішує задачу факторизації чисел. Загальна часова складність удосконаленого алгоритму, яка визначається сумою складностей основних операцій, становить $O\left(log_2 n \cdot \left(log_2 n + \frac{n}{2} + k \cdot log_2 n\right) + log_2 \frac{n}{2}\right)$.

Приклади застосування алгоритмів пошуку НСД.

Нехай потрібно обчислити НСД(3843, 1449).

1. Стандартний алгоритм Евкліда:

3843=1449·1+945

1449=945·1+504

945=504·1+441

504=441·1+63

441=63·7+0

Отже, НСД(3843, 1449)=63.

3. Алгоритм Евкліда в розмежованій системі числення зручно представити у вигляді таблиці 3.

Таблиця 3 – Алгоритм Евкліда в розмежованій системі числення

1	3843	1	1	1	1	0	0	0	0	0	0	1	1
2	$2^i \bmod 1449$	599	1024	512	256	128	64	32	16	8	4	2	1
3	1449		1	0	1	1	0	1	0	1	0	0	1
4	$2^i \bmod 945$		79	512	256	128	64	32	16	8	4	2	1
5	945			1	1	1	0	1	1	0	0	0	1
6	$2^i \bmod 504$			8	256	128	64	32	16	8	4	2	1
7	504				1	1	1	1	1	1	0	0	0
8	$2^i \bmod 441$				256	128	64	32	16	8	4	2	1
9	441				1	1	0	1	1	1	0	0	1
10	$2^i \bmod 63$				4	2	1	32	16	8	4	2	1

З рядка 2 видно, що (599+1024+512+256+2+1)mod 1449=945.

З рядка 4: (79+256+128+32+8+1)mod 945=504.

З рядка 6: (8+256+128+32+16+1)mod 504=441.

З рядка 8: (256+128+64+32+16+8)mod 441=63.

З рядка 10: (4+2+32+16+8+1)mod 63=0.

Таким чином можна отримати НСД, уникнувши громіздкої операції ділення.

3. Пошук НСД в базисі Крестенсона.

Дану задачу також зручно представити у вигляді таблиці 4, врахувавши, що $\sqrt{1449} \approx 38$.

Таблиця 4 – Знаходження залишків по простих модулях.

1		2048	1024	512	256	128	64	32	16	8	4	2	1
2	3843	1	1	1	1	0	0	0	0	0	0	1	1
3	1449		1	0	1	1	0	1	0	1	0	0	1

4	$2^i \bmod 2$	0	0	0	0	0	0	0	0	0	0	0	1
5	$2^i \bmod 3$	2	1	2	1	2	1	2	1	2	1	2	1
6	$2^i \bmod 5$	3	4	2	1	3	4	2	1	3	4	2	1
7	$2^i \bmod 7$	4	2	1	4	2	1	4	2	1	4	2	1
8	$2^i \bmod 11$	2	1	6	3	7	9	10	5	8	4	2	1
9	$2^i \bmod 13$	7	10	5	9	11	12	6	3	8	4	2	1
10	$2^i \bmod 17$	8	4	2	1	9	13	15	16	8	4	2	1
11	$2^i \bmod 19$	15	17	18	9	14	7	13	16	8	4	2	1
12	$2^i \bmod 23$	1	12	6	3	13	18	9	16	8	4	2	1
13	$2^i \bmod 29$	18	9	19	24	12	6	3	16	8	4	2	1
14	$2^i \bmod 31$	2	1	16	8	4	2	1	16	8	4	2	1
15	$2^i \bmod 37$	13	25	31	34	17	27	32	16	8	4	2	1
16	$2^i \bmod 9$	5	7	8	4	2	1	5	7	8	4	2	1
17	$2^i \bmod 27$	23	25	26	13	20	10	5	16	8	4	2	1

З таблиці 4 шукаються залишки по простих модулях.

Рядок4: 3843 mod 2=1; 1449 mod 2=1;

Рядок 5: 3843 mod 3=(2+1+2+1+2+1) mod 3=0; 1449 mod 3=(1+1+2+2+2+1) mod 3=0;

Рядок 6: 3843 mod 5=(3+4+2+1+2+1) mod 5=3; 1449 mod 3=(4+1+3+2+3+1) mod 5=4;

Рядок 7: 3843 mod 7=(4+2+1+4+2+1) mod 7=0; 1449 mod 7=(2+4+2+4+1+1) mod 7=0;

Рядок 8: 3843 mod 11=(2+1+6+3+2+1) mod 11=4; 1449 mod 11=(1+3+7+10+8+1) mod 3=8;

Рядок 9: 3843 mod 13=(7+10+5+9+2+1) mod 13=8; 1449 mod 3=(10+9+11+6+8+1) mod 13=6;

Рядок 10: 3843 mod 17=(8+4+2+1+2+1) mod 17=1; 1449 mod 17=(4+1+9+15+8+1) mod 17=4;

Рядок 11: 3843 mod 19=(15+17+18+9+2+1) mod 19=5; 1449 mod 19=(17+9+14+13+8+1) mod 19=5;

Рядок 12: 3843 mod 23=(1+12+6+3+2+1) mod 23=2; 1449 mod 23=(12+3+13+9+8+1) mod 23=0;

Рядок 13: 3843 mod 29=(18+9+19+24+2+1) mod 29=15; 1449 mod 29=(9+24+12+3+8+1) mod 29=28;

Рядок 14: 3843 mod 31=(2+1+16+8+2+1) mod 31=30; 1449 mod 31=(1+8+4+1+8+1) mod 31=23;

Рядок 15: 3843 mod 37=(13+25+31+34+2+1) mod 37=32; 1449 mod 37=(25+34+17+32+8+1) mod 37=6.

Дані розрахунки показують, що спільними простими дільниками є числа 3 та 7. Для знаходження НСД потрібно перевірити їх степені:

Рядок 16: 3843 mod 3^2=(5+7+8+4+2+1) mod 3^2=0; 1449 mod 3^2 =(7+4+2+5+8+1) mod 3^2=0;

Рядок 17: 3843 mod 3^3=(23+25+26+13+2+1) mod 3^3=9; 1449 mod 3^3 =(25+13+20+5+8+1) mod 3^3=18.

Число 7^2=49>38 і його можна не перевіряти. Отже, НСД(3843, 1449)= 3^2·7=63. Даний метод дозволяє провести факторизацію чисел, наприклад 1449=3^2·7·23. Крім того, обчислення можна виконувати паралельно по різних модулях.

4. Удосконалений алгоритм пошуку НСД в базисі Крестенсона.

Будується таблицю 5.

Таблиця 5 – Знаходження залишків в удосконаленому алгоритмі.

1		2048	1024	512	256	128	64	32	16	8	4	2	1
2	3843	1	1	1	1	0	0	0	0	0	0	1	1
3	1449		1	0	1	1	0	1	0	1	0	0	1
4	2^imod2	0	0	0	0	0	0	0	0	0	0	0	1
5	2^imod3	2	1	2	1	2	1	2	1	2	1	2	1
6	2^imod9	5	7	8	4	2	1	5	7	8	4	2	1
7	2^imod27	23	25	26	13	20	10	5	16	8	4	2	1
8	2^imod45	23	34	17	31	38	19	32	16	8	4	2	1
9	2^imod63	32	16	8	4	2	1	32	16	8	4	2	1

10	$2^i \bmod 441$	284	142	71	256	128	64	32	16	8	4	2	1
11	$2^i \bmod 693$	662	331	512	256	128	64	32	16	8	4	2	1

Аналізуємо таблицю 9.5.

Рядок 4: 3843 mod 2=1; 1449 mod 2=1;

Рядок 5: 3843 mod 3=(2+1+2+1+2+1) mod 3=0; 1449 mod 3=(1+1+2+2+2+1) mod 3=0;

Рядок 6: 3843 mod 9=(5+7+8+4+2+1) mod 9=0; 1449 mod 9 =(7+4+2+5+8+1) mod 9=0;

Рядок 7: 3843 mod 27=(23+25+26+13+2+1) mod 27=9; 1449 mod 27 =(25+13+20+5+8+1) mod 27=18;

Рядок 8: 3843 mod 45=(23+34+17+31+2+1) mod 45=18; 1449 mod 45=(34+31+38+32+8+1) mod 3=8;

Рядок 9: 3843 mod 63=(32+16+8+4+2+1) mod 63=0; 1449 mod 63=(16+4+2+32+8+1) mod 63=0;

Рядок 10: 3843 mod 441=(284+142+71+256+2+1) mod 441=315; 1449 mod 441=(142+256+128+32+8+1) mod 441=126;

Рядок 11: 3843 mod 693=(662+331+512+256+2+1) mod 693=378; 1449 mod 693=(331+256+128+32+8+1) mod 693=63.

З розрахунків також випливає, що НСД(3843, 1449)=63.

Крім того, зазначимо, що в двох останніх алгоритмах не обов'язково шукати залишки від обох чисел a та b. Досить знайти залишки від меншого числа і тільки при їх рівності 0 перевіряти друге число.

Складність запропонованих алгоритмів визначається часовою складністю наступних операцій:

1) знаходженні залишків a_j, b_j чисел X, Y по простих модулях $p_j^{m_j}$ для яких виконується умова $a_j = b_j = 0$.

2) обчислення добутку модулів $Z = НСД(X,Y) = \prod_{j=1}^{k} p_j^{m_j}$.

В таблиці 6 подано оцінки часової складності основних операцій алгоритму пошуку НСД в базисі Крестенсона, що дозволяє зробити

порівняльний аналіз з вищенаведеними алгоритмами пошуку найбільшого спільного дільника.

Таблиця 6 – Часова складність основних операцій алгоритму пошуку НСД в базисі Крестенсона та його удосконалення.

	Основні операції	Часова складність
.	$p_j^{m_j}$	$O\left(\log_2 n \cdot \left(\log_2 n + \frac{n}{2}\right)\right)$
.	$a_j^{(m)} = res\left(\sum_{i=1}^{n-1} a_{ij} (\mathrm{mod}\, p_j^m)\right)$ $b_j^{(m)} = res\left(\sum_{i=1}^{n-1} b_{ij} (\mathrm{mod}\, p_j^m)\right)$	$O\left(\log_2 n/2\right)$
.	$Z = \prod_{j=1}^{k} p_j^{m_j}$	$O(k \cdot \log_2 n)$

де k - кількість модулів для яких виконується умова $a_j = b_j = 0$.

З врахуванням даних табл. 9.6, загальна часова складність запропонованого алгоритму пошуку НСД в базисі Крестенсона, та його удосконалення буде визначатися сумою складностей основних операцій, а саме: $O\left(\log_2 n \cdot \left(\log_2 n + \frac{n}{2} + k \cdot \log_2 n\right) + n \cdot \log_2 n/2\right)$ і $O3\left(\log_2 n\left(\log_2 n + k \cdot \log_2 n + \frac{n}{2}\right) + \frac{n}{2} \cdot \log_2 \frac{n}{2}\right)$ відповідно. На рис. 9.1 показані графіки які характеризують складності існуючого та запропонованих алгоритмів в залежності від розрядності компонентів Z.

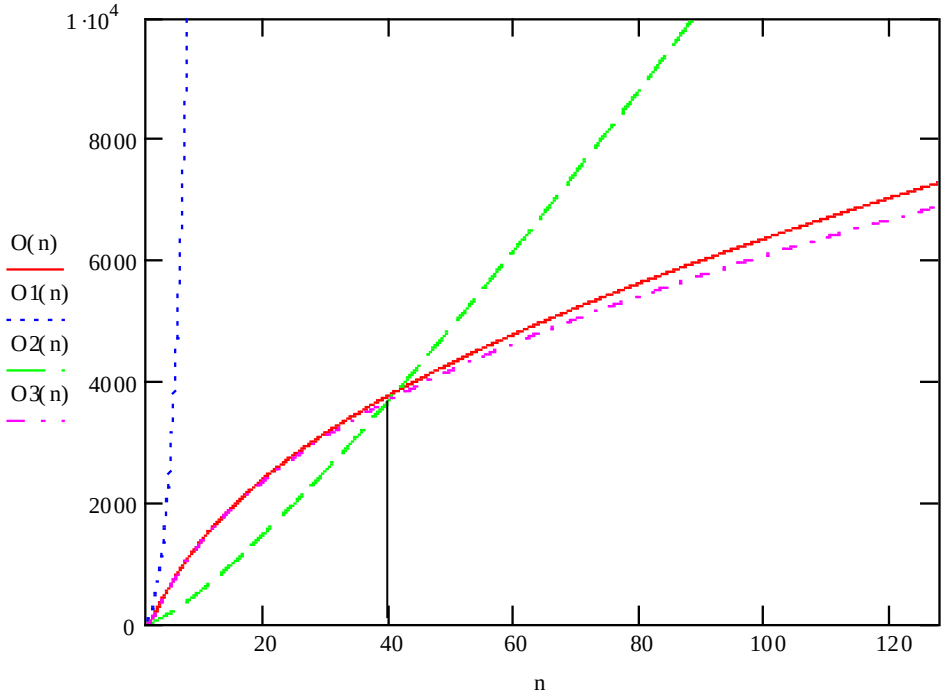


Рисунок 9.1 – Складності алгоритмів пошуку НСД(X, Y) , $O(n)$ - складність алгоритму пошуку НСД в базисі Крестенсона, $O1(n)$ - алгоритму Евкліда, $O2(n)$ - удосконалення реалізації алгоритму Евкліда з використанням розмежованої системи числення Рademахера – Крестенсона, $O3(n)$ - удосконалений алгоритм пошуку НСД в базисі Крестенсона.

Результати досліджень показали, що для пошуку НСД двох чисел існуючий алгоритм Евкліда, який традиційно використовується для пошуку НСД для чисел великої розрядності при сучасному рівні комп'ютерної техніки стає практично нездійсненний. Чисельний експеримент оцінки складностей запропонованих алгоритмів пошуку НСД показує, що в діапазоні двійкових розрядів від 0 до 40 бітів, слід використовувати удосконалення реалізації алгоритму Евкліда з використанням розмежованої системи числення Рademахера – Крестенсона, а при збільшенні розрядності чисел потрібно застосовувати алгоритм алгоритм пошуку НСД в базисі Крестенсона та його удосконалення.

Лекція 9.

Китайська теорема про залишки

Перетворення Китайської теореми про залишки (КТЗ) є фундаментальною основою вирішення широкого класу задач теорії чисел, а також прикладних задач інженерії та інформатики.

Незважаючи на свою простоту та древню історію, КТЗ продовжує представляти себе у новому світлі і відкривати нові перспективи свого застосування, особливо у математиці, інформатиці (машинна арифметика), криптографії тощо. Побудова непозиційної системи числення в часових системах (системи залишкових класів) для виконання операцій з великими числами, дискретне перетворення Фур'є, генерування таємних ключів в асиметричних криптосистемах, зв'язок з класичною поліноміальною інтерполяційною теорією, багатовимірні обчислення, можливість зведення вивчення кільця лишків за модулем m (де m – довільне ціле число) до вивчення кільця лишків за модулем p^s (p – просте число), дослідження алгебраїчних кілець, можливість арифметичної самокорекції кодів та розпаралелення обчислень, визначення послідовності великого числа зразків ДНК – ось далеко не повний перелік сучасного застосування КТЗ.

КТЗ є одним з найдавнішим, але важливим часовим алгоритмом. Ще в першому столітті нашої ери китайський математик Сунь–Цзи придумав загадку, якою було покладено початок модулярній арифметиці: знайти число, яке при діленні на 3 дасть в остачі 2, на 5 – 3, на 7 – 2. Крім того, він показав у частковому випадку еквівалентність розв'язку системи модулярних рівнянь і розв'язку одного модулярного рівняння.

Протягом майже двох тисяч років КТЗ постійно вдосконалювалася та розвивалася. Зокрема, в XIII столітті інший китайський математик Цань Цзю–шао розв'язав наведену вище задачу. У XVIII столітті німецький математик Л.Ейлер навів загальне формулювання та доведення КТЗ, а К.–Ф. Гаус істотно розвинув його в своїх знаменитих „Арифметичних дослідженнях".

І, нарешті, в середині XX століття чеські учені М.Валах та А.Свобода запропонували використати древню китайську ідею на новому технічному

рівні, створивши перші модулярні електронно–часові машини „Епос” та „Епос–2”. Їх ідеї підтримали радянські та українські вчені Ф.Лукін, І.Акушський, Д.Юдіцький, Є.Адріанов, В.Амербаєв, Я. Николайчук та інші.

Слід зазначити, що на даний час існує декілька еквівалентних формулювань КТЗ. Найбільш поширене з них таке: якщо натуральні числа p_1, p_2, …, p_k попарно взаємно прості, то для будь–яких цілих r_1, r_2, …, r_k, таких що $0 \le r_i < p_i$ існує число N, яке при діленні на p_i дає залишок r_i при всіх $i=1, 2, ..., k$; більше того, якщо існує два таких числа N_1 та N_2, то $N_1 \ mod \ P = N_2 \ mod \ P$, де $P = \prod_{i=1}^{k} p_i$.

Дану теорему можна можна представити у вигляді системи порівнянь:

$$\begin{cases} N \bmod p_1 = r_1 \\ N \bmod p_2 = r_2 \\ \ldots\ldots\ldots\ldots \\ N \bmod p_i = r_i \\ \ldots\ldots\ldots\ldots \\ N \bmod p_k = r_k . \end{cases} \qquad (9.5)$$

Шукане число обчислюється за формулою:

$$N = \left(\sum_{i=1}^{k} M_i m_i r_i \right) \bmod P , \qquad (9.6)$$

де $M_i = \dfrac{P}{p_i} = p_1 p_2 \ldots p_{i-1} p_{i+1} \ldots p_k$, $m_i = M_i^{-1} \bmod p_i$.

Відмітимо, що на даний час відомі три способи пошуку оберненого елемента m_i:

1) перевірка шляхом послідовної підстановки чисел натурального ряду у формулу, поки не буде виконуватись умова $M_i m_i \bmod p_i = 1$;

2) використовуючи функцію Ейлера, можна знайти $m_i = M_i^{-1} \bmod p_i = M_i^{\varphi(p_i - 1)} \bmod p_i$;

3) за допомогою розширеного алгоритму Евкліда.

Однак кожен з цих способів характеризується значною часовою складністю при виконанні ділень з остачею, піднесення до степеня, знаходженні функції Ейлера (факторизації p_i). Причому всі ці операції повинні виконуватися над дуже великими числами, що приводить до переповнення розрядної сітки сучасних потужних часових засобів.

Професором Николайчуком Я.М. була запропонована досконала форма системи залишкових класів, у якій підбір модулів такий, що $M_i \bmod p_i = 1, m_i = 1$. В подальшому було розвинуто дану теорію та розроблено її модифікований варіант, коли $M_i \bmod p_i = \pm 1, m_i = \pm 1$, тобто відповідний підбір модулів дозволяє уникнути процедури знаходження оберненого елемента. Недолік даного методу полягає в тому, що не завжди є можливість вибору відповідної системи модулів.

2. Теоретичні основи алгоритмів перетворення КТЗ в базисі Радемахера–Крестенсона.

Для спрощення розглянемо два взаємно прості модулі $p_1 < p_2$. Нехай потрібно знайти число N, яке при діленні на p_1 дає залишок r_1, а при діленні на p_2 – залишок r_2, що еквівалентно такій системі порівнянь:

$$\begin{cases} N \bmod p_1 = r_1 \\ N \bmod p_2 = r_2. \end{cases} \qquad (9.7)$$

Розв’язок (9.7) можна представити у такому вигляді:

$$N = \left(r_1 p_2 \left(p_2^{-1} \bmod p_1 \right) + r_2 p_1 \left(p_1^{-1} \bmod p_2 \right) \right) \bmod p_1 p_2. \qquad (9.8)$$

Для знаходження $p_2^{-1} \bmod p_1$ представимо p_2 у двійковій формі: $p_2 = a_{n-1} 2^{n-1} + a_i \cdot 2^i + \ldots + a_1 \cdot 2^1 + + a_0 \cdot 2^0$, де $a_i = 0,1$, і сформуємо таблицю 9.7.

Щоб знайти елемент p_{2i} необхідно попередній елемент p_{2i-1} домножити на 2 (дописати в кінці 0 у двійковому записі) і порівняти з модулем p_1. остаточна формула для p_{2i} матиме такий вигляд:

Таблиця 9.7

Знаходження залишків степенів двійки

2^{n-1}	2^{n-2}	…	2^i	…	2	1
a_{n-1}	a_{n-2}	…	a_i	…	a_1	a_0
$p_{2\,n-1}$	$p_{2\,n-2}$	…	$p_{2\,i}$	…	$p_{2\,1}$	$p_{2\,0}$

$$p_{2i} = \begin{cases} 2 \cdot p_{2\ i-1},\ 2 \cdot p_{2\ i-1} < p_1 \\ 2 \cdot p_{2\ i-1} - p_1,\ 2 \cdot p_{2\ i-1} \geq p_1. \end{cases} \qquad (9.9)$$

Отже, уникнувши громіздкої операції ділення, знаходимо залишок $p_3=p_2\ mod\ p_1$. Він буде дорівнювати сумі тих p_{2i}, для яких відповідні $a_i=1$. Тоді $p_2^{-1} \bmod p_1 = p_3^{-1} \bmod p_1$.

Для знаходження оберненого елемента знову ж шукаємо залишок $p_1\ mod\ p_3=p_{10}$. Оскільки $p_{10}\neq 0$, то далі виконується така послідовність кроків: $(p_1+1) mod\ p_3=(p_{10}+1) mod\ p_3=p_{11}$; $(2p_1+1) mod\ p_3=(p_{11}+p_1) mod\ p_3=p_{12}$; …. ; $(i \cdot p_1+1) mod\ p_3=(p_{1i-1}+p_1) mod\ p_3=p_{1i}$; … . Описану послідовність продовжуємо до тих пір, поки p_{1i} не стане рівним нулю. Зазначимо, що процедура знаходження p_{1i} аналогічна визначенню залишку p_3.

Обернений елемент $p_2^{-1}\ mod\ p_1$ дорівнюватиме результату ділення $(i \cdot p_1+1)$ на p_3. Для уникнення цієї громіздкої операції потрібно описаним вище методом знайти залишки $b_i=(i \cdot p_1+1)\ mod\ p_3 \cdot q_i^s$, де q_i пробігає послідовність простих чисел, $p_3 \cdot q_i^s < (i \cdot p_1+1)$, $s=1, 2, \ldots$, причому s збільшується на 1, коли $b_i=0$. Шукане обернене число p_2^{-1} буде дорівнювати добутку тих q_i^s, для яких відповідні $b_i=0$.

За аналогічним алгоритмом шукається $p_1^{-1}\ mod\ p_2$.

Наступним кроком є обчислення добутків трьох множників за модулем P у кожному доданку (9.9). Операцію множення пропонуємо виконати матричним методом, що істотно зменшує часову складність.

Розглянемо два числа $x=x_{n-1}2^{n-1}+\ldots+x_i2^i+\ldots+x_12+x_0$ та $y=y_{n-1}2^{n-1}+\ldots+y_j2^j+\ldots++y_12+y_0$, де $x_i, y_j=0, 1$, n–розрядність модуля P. Для знаходження результату їх множення за модулем P побудуємо матрицю, представлену в таблиці 9.8, де $c_{ij}=2^{i+j} mod\ P$.

Таблиця 9.8 – Матриця для множення двох n–розрядних двійкових чисел

	b_{n-1}	…	b_j	…	b_1	b_0
a_{n-1}	$c_{n-1\ n-1}$	…	$c_{n-1\ j}$	…	$c_{n-1\ 1}$	$c_{n-1\ 0}$
…	…	…	…	…	…	…
a_i	$c_{i\ n-1}$	…	c_{ij}	…	c_{i1}	c_{i0}
…	…	…	…	…	…	…
a_1	$c_{1\ n-1}$	…	c_{1j}	…	c_{11}	c_{10}
a_0	$c_{0\ n-1}$	…	c_{0j}	…	c_{01}	c_{00}

Добуток чисел x та$\cdot y$ отримуємо за формулою:

$$x \cdot y = \left(\sum_{m,k=1}^{n-1} c_{mk} \right) \mathrm{mod}\ \ P, \qquad (9.10)$$

де x_m, $y_k=1$, тобто c_{mk} знаходиться на перетині стовбця та рядка, для яких відповідні x_i та y_j дорівнюють 1.

Останнім кроком знаходження шуканого числа N є визначення суми двох чисел за модулем P.

3. Застосування запропонованих алгоритмів.

Розглянемо приклад. Нехай потрібно знайти число N, яке при діленні на $p_1=43$ дає остачу $r_1=10$, а при діленні на $p_2=209$ – остачу r_2 =100 ($P=209\cdot43=8987$).

Знайдемо $p_3=209\ mod\ 43$ матричним методом, представленим у таблиці 9.9.

Таблиця 8.9 – Знаходження залишку за модулем

2^i	128	64	32	16	8	4	2	1
209	1	1	0	1	0	0	0	1
$2^i mod$ 43	42	21	32	16	8	4	2	1

Отже, p_3=(42+21+16+1) *mod* 43=37.

Далі знаходимо $p_3^{-1} mod 43$=$37^{-1} mod$ 43. Для цього шукаємо 43 *mod* 37=6, додаємо 1 і послідовно додаємо 6, поки в результаті додавання за *mod* 37 не буде 0. Представимо це у вигляді таблиці 9.10.

Таблиця 9.10 – Пошук оберненого елемента за модулем

i	0	1	2	3	4	5	6
p_{1i}	6	7	13	19	25	31	0

Звідси число K_1=6·43+1, яке націло ділиться на 37. Добуток 6·43 знайдемо також матричним методом, представленим у таблиці 9.11, записавши обидва множники у двійковій формі: 6=$(110)_2$; 43=$(101011)_2$.

Таблиця 9.11 – Множення двох n–розрядних двійкових чисел

	0	0	0	1	1	0
1	1024	512	256	128	64	32
0	512	256	128	64	32	16
1	256	128	64	32	16	8
0	128	64	32	16	8	4
1	64	32	16	8	4	2
1	32	16	8	4	2	1

Отже, K_1=6·43+1=(128+64+32+16+8+4+4+2)+1=259. Діленням можна знайти, що $p_3^{-1} \bmod p_1$=$37^{-1} mod$ 43=259:37=7. Матричним методом це представлено в таблиці 9.12 (не перевіряючи парного простого числа 2).

Таблиця 9.12–Пошук оберненого елемента $37^{-1} mod$ 43

2^i	256	128	64	32	16	8	4	2	1	

259	1	0	0	0	0	0	0	1	1	
2^i*mod* 3·37	34	17	64	32	16	8	4	2	1	(34+2+1)*mod* 111=37
2^i*mod* 5·37	71	128	64	32	16	8	4	2	1	(71+2+1)*mod* 185=74
2^i*mod* 7·37	256	128	64	32	16	8	4	2	1	(256+2+1)*mod* 259=0

Звідси видно, що $p_3^{-1} \bmod p_1 = 37^{-1} mod\ 43 = 7$. Далі шукається $p_1^{-1} \bmod p_2 = 43^{-1} mod\ 209$. Вище було знайдено, що 209 *mod* 43=37. Будуться таблиця 9.13.

Таблиця 9.13 – Пошук оберненого елемента $43^{-1} mod\ 209$ в розмежованому базисі

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
p_{2i}	37	38	32	26	20	14	8	2	39	33	27	21	15	9	3	40	34	28	22
i	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
p_{2i}	16	10	4	41	35	29	23	17	11	5	42	36	30	24	18	12	6	0	

Тоді $K_2 = 36 \cdot 209 + 1 = 7525$ і $p_1^{-1} \bmod p_2 = 43^{-1} mod\ 209 = 7525 : 43 = 175$. Матричним методом виходить аналогічний результат, представлений у таблиці 9.14.

Таблиця 9.14 – Пошук оберненого елемента $43^{-1} mod 209$ матричним методом

2^i	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	
7525	1	1	1	0	1	0	1	1	0	0	1	0	1	
2^i*mod* 3·43	97	113	121	125	127	128	64	32	16	8	4	2	1	(97+113+121+ +127+64+32+ +4+1) *mod* 129=45
2^i*mod* 5·43	11	113	164	82	41	128	64	32	16	8	4	2	1	(11+113+164 +41+64+32+ +4+1)*mod* 215=0
2^i*mod* 5^2·43	871	973	1024	512	256	128	64	32	16	8	4	2	1	(871+973+1024+ +256+64+32+4+ +1)*mod* 1075=0
2^i*mod* 5^3·43	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	(4096+2048+1024+ +256+64+32+4+

														+1)*mod* 5375=2150
$2^i mod$ $5^2 \cdot 7 \cdot 43$	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	(4096+2048+1024+ +256+64+32+4+ +1)*mod* 7525=0

Отже, $p_1^{-1} \bmod p_2 = 43^{-1} mod\ 209 = 5^2 \cdot 7 = 175$. Звідси видно, що шукане число: $N = (209 \cdot 7 \cdot 10 + +43 \cdot 175 \cdot 100)\ mod\ 8987 = 3235$.

4. Оцінка та порівняльний аналіз часових складностей відомих та запропонованих алгоритмів.

При перетвореннях згідно КТЗ використовуються такі основні модульні операції:

1) знаходження оберненого елемента;

2) знаходження залишків;

3) операції множення та додавання.

Тому при визначенні часових складностей відомого та запропонованого алгоритмів, які дозволяють виконувати перетворення КТЗ, потрібно враховувати складності вищезазначених операцій, наведені у таблиці 9.15.

Враховуючи табличні дані, часова складність Китайської теореми про залишки з використанням ТЧБ Радемахера-Крестенсона становить $O\left(\left(\log_2 k \cdot (2 \cdot \log_2{}^2\, n + n\right) + \frac{n^2 \cdot k}{2} + \left(\log_2 n/2\right)\right)$, а з використанням класичного алгоритму – $O1(37k \cdot n^2 + 53{,}5k \cdot n + 17{,}5k + n^2 + 3n + 1))$.

Таблиця 9.15 – Часові складності основних операцій КТЗ

№	Основні операції	Часова складність операцій у запропонованому алгоритмі	Часова складність операцій у класичному алгоритмі
1.	Пошук оберненого елемента	$O\left(\frac{n^2 \cdot k}{2} \cdot\right)$	$O1(17{,}5k \cdot ((n+1)^2 + n^2 + n)$
2.	Пошук залишків	$O\left(log_2 n/2\right)$	$O1((n+1)^2 + n)$
3.		$O\left(log_2 k \cdot (2 \cdot log_2{}^2\, n + n\right)$	$O1(k \cdot (2n^2 + n))$

де k - кількість взаємно простих модулів $n = n_1 * n_2 * \ \ldots\ * n_k$.

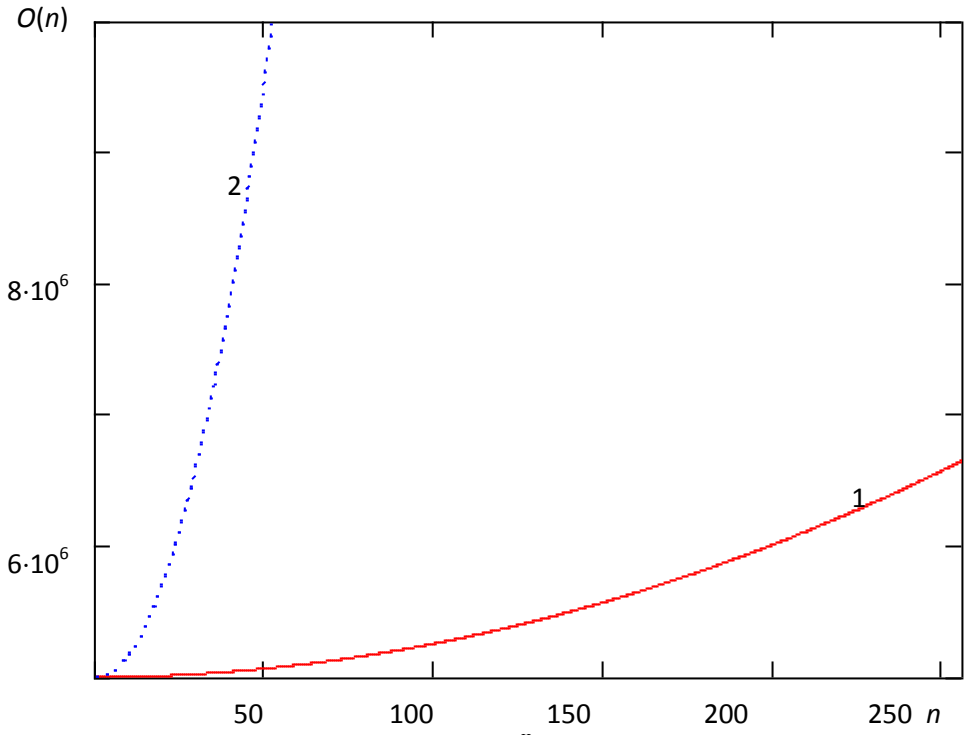


Рисунок 9.2 – Графіки залежності часових складностей від розрядності чисел *n* методом з використанням ТЧБ Радемахера – Крестенсона (1) та класичним (2).

На рисунку 9.2 показано графіки залежності часових складностей від розрядності чисел *n*. З рисунка видно, що використання запропонованого алгоритму, який грунтується на використанні теоретико-числового базису Крестенсона, дозволяє істотно зменшити часову складність КТЗ відносно класичного.

Лекція 10

Високопродуктивні алгоритми множення багаторозрядних чисел в базисі Крестенсона

Однією з найважливіших операцій у всіх асиметричних алгоритмах шифрування інформаційних потоків є модулярне множення багато розрядних чисел a та b розміру n. Для підвищення ефективності обчислення $a \cdot b (\mathrm{mod}\ p)$ багатьма авторами запропоновано різні підходи, алгоритми, але в основному з використанням десяткової системи числення з часовою складністю $O(n^2)$, що значно збільшує час виконання програми. Одним з підходів щодо підвищення швидкодії знаходження результату модулярного множення є метод Карабуци, але він практично не використовують через складність його реалізації. В алгоритмі Шенхаге – Штрассена з використанням швидкого перетворення Фур'є і потребує $O(nlog(n)log(log(n))$ бітових операцій.

Тому розробка ефективного алгоритму з використанням теоретико-числового базису Крестенсона для зменшення часової складності формування, збільшення швидкодії на основі матричних моделей та розробка спеціалізованих програмно-апаратних засобів є актуальною задачею.

Операції модулярного множення в базисі Радемахера лежать в основі алгоритмів електронного цифрового підпису, криптографічних протоколів, розв'язування задач часової, прикладної та дискретної математики. Визначення стійкості еліптичних кривих методом пошуку їх порядку за допомогою алгоритму Шуфа тощо. Існуючі алгоритми (стандартний, швидкого множення, Blakey, Монтгомері, бінарні і т.д.) характеризуються значною часовою складністю. Запропонований алгоритм модулярного множення в базисі Крестенсона за допомогою матричних обчислень, дозволить зменшити часову складність.

Відомо, що в базисі Крестенсона будь–яке ціле десяткове число N представляється у вигляді набору найменших невід'ємних залишків від його ділення на фіксовані цілі додатні взаємно прості модулі p_i, причому

$0 \le N \le \prod_{i=1}^{n} p_i - 1$. Зворотнє перетворення в десяткову систему числення є набагато складнішим. Оскільки операції в комп'ютерних системах виконуються в базисі Радемахера, то постає питання в прямому переході з базису Крестенсона в базис Радемахера і навпаки, що ефективно виконується за допомогою принципової розмежованої форми системи числення в базисах Радемахера-Крестенсона.

Розглянемо два n–розрядних числа $a=a_{n-1}2^{n-1}+\ldots+ +a_i2^i+\ldots+ a_12+a_0$ та $b=b_{n-1}2^{n-1}+\ldots+ +b_i2^j+\ldots+ b_12+b_0$, де a_i, b_j=0, 1, n–розрядність модуля p. Для знаходження результату їх множення за модулем p побудуємо матрицю, представлену в таблиці 9.16, де $c_{ij}=2^{i+j} \bmod p$.

Добуток чисел a та·b отримуємо за формулою:

$$a \cdot b = \left(\sum_{m,k=1}^{n-1} c_{mk} \right) \bmod p, \qquad (9.10)$$

де a_m, b_k=1, тобто c_{mk} знаходиться на перетині стовбця та рядка, для яких відповідні a_i та b_j дорівнюють 1

Таблиця 9.16 – Матриця множення в базисі Радемахера–Крестенсона

	b_{n-1}	…	b_j	…	b_1	b_0
a_{n-1}	$c_{n-1\ n-1}$	…	$c_{n-1\ j}$	…	$c_{n-1\ 1}$	$c_{n-1\ 0}$
…	…	…	…	…	…	…
a_i	$c_{i\ n-1}$	…	c_{ij}	…	c_{i1}	c_{i0}
…	…	…	…	…	…	…
a_1	$c_{1\ n-1}$	…	c_{1j}	…	c_{11}	c_{10}
a_0	$c_{0\ n-1}$	…	c_{0j}	…	c_{01}	c_{00}

Модулярне множення здійснюється згідно алгоритму:

Таким чином, отриманий новий алгоритм заміни операції множення, яка має квадратичну часову складність $O1(n) = n^2$, операцією додавання з логарифмічною складністю $O2(n) = \begin{cases} (\log_2 n)^2, \text{ якщо } n < 256 \\ n \cdot (\log_2 n), \text{ в інших випадках} \end{cases}$.

Результати дослідження ефективності запропонованого алгоритму приведена на рисунку 9.3.

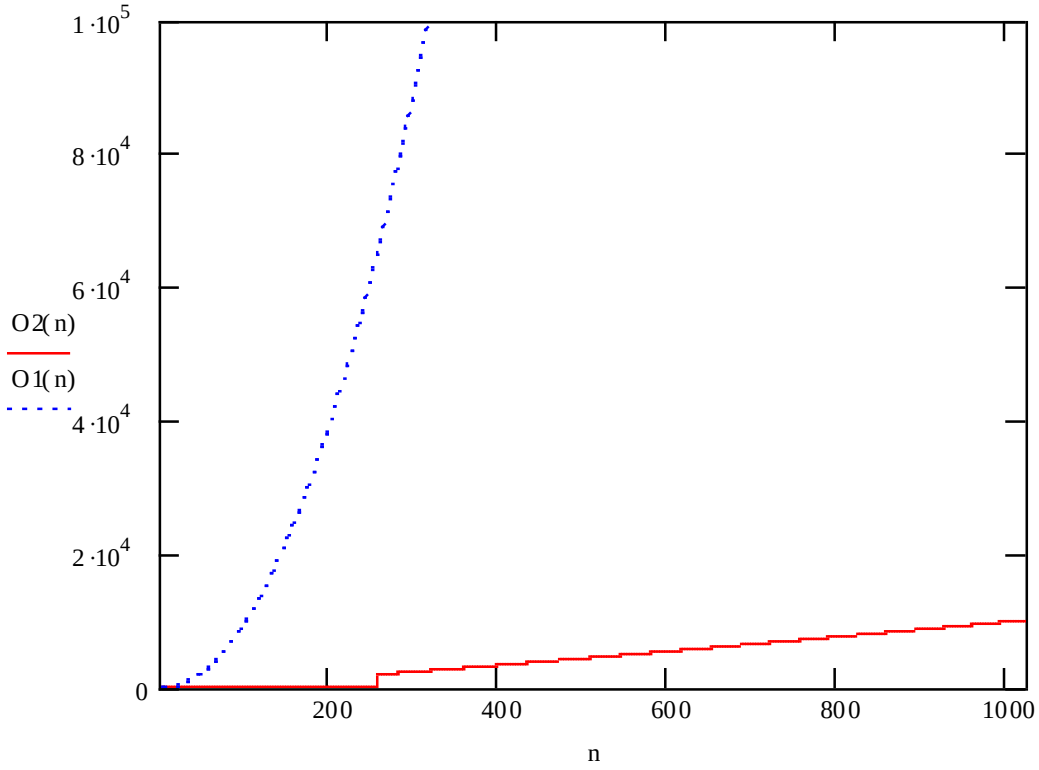


Рисунок 9.3. – Складність операції модулярного множення

З рисунка 9.3 видно, що при зростанні розрядності компонентів дискретного логарифма n>256 відомий алгоритм не може бути фізично реалізований на даний час з використанням сучасної мікропроцесорної техніки, особливо з врахуванням наступної операції піднесення до степеня, яка включає операцію множення. А запропонований алгоритм з використанням базису Крестенсона дозволяє реалізувати можливості суттєвого підвищення захисту інформаційних потоків від несанкціонованого доступу, які характеризуються ознаками незворотності, як показано в роботах Николайчука Я.М., повинно бути практично нездійсненна, що строго математично не доведено.

Ефективність запропонованого алгоритму модулярного множення буде

$$EЗ(n) = \begin{cases} \dfrac{n^2}{(\log_2 n)^2}, & якщо\ n < 256 \\ \dfrac{n^2}{n \cdot \log_2 n}, & в\ інших\ випадках. \end{cases}$$

(рис.9.4)., як співвідношення часових складностей.

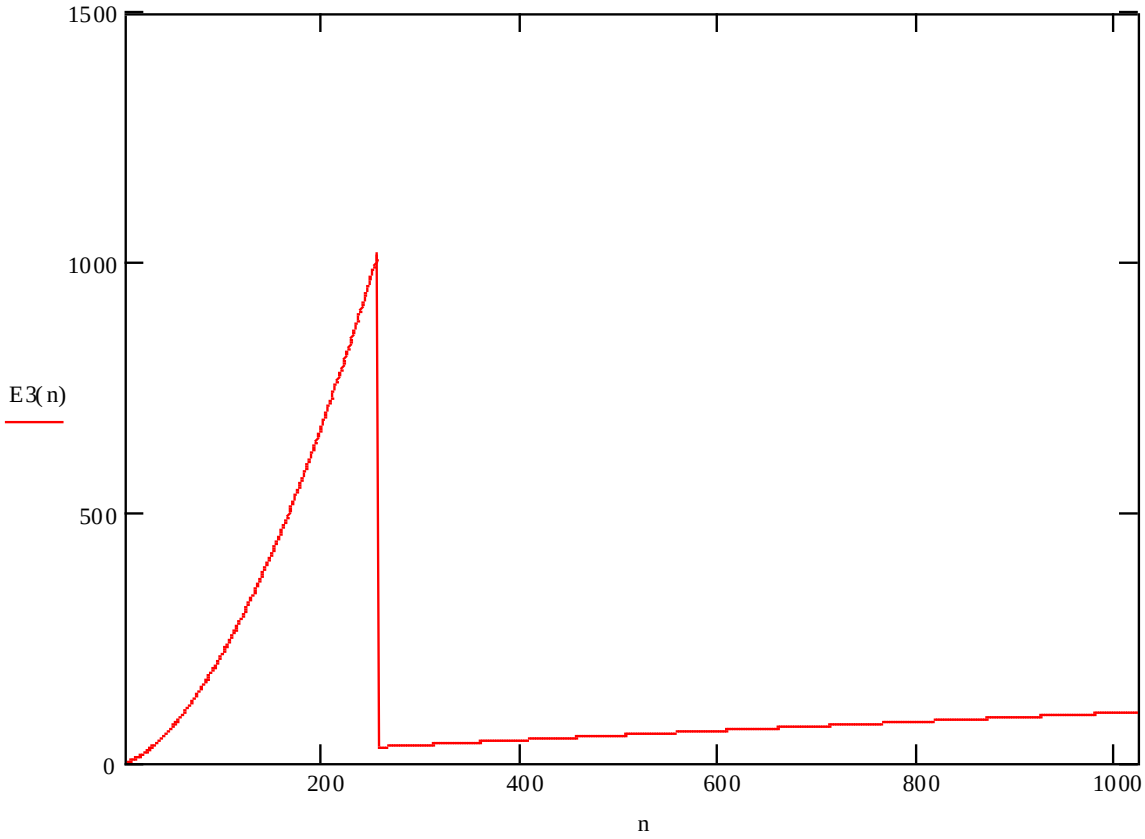


Рисунок 9.4 – Ефективність алгоритму модулярного множення розмірності n.

Результати чисельного експеременту показують, що при розмірності чисел від 0 до 256 бітів ефективність стрімко зростає, за рахунок переходу від двовимірного базису Радемахера, складність операції модулярного множення в якому мають квадратичну залежність, до розмежованої системи числення Радемахера-Крестенсона з логарифмічною складністю. А в діапазоні від 256 до 1024 біти в розмежованій системі числення складність операції множення буде лінійно-логарифмічною, тому що при реалізації запропонованого алгоритму операція додавання чисел великої розрядності більше 256 біт, яка замінює

операцію множення в відомих алгоритмах, здійснюється не за один такт, як при меших розрядах, а за декілька тактів.

Отже, з використанням базису Крестенсона суттєво збільшуються перспективи щодо розробки систем з високим рівнем захисту інформаційних потоків та дозволяє пришвидшити час виконання алгоритму модулярного множення.

Лекція 11

Розробка і дослідження алгоритмів піднесення до високих показників степенів у розмежованій системі числення базису Крестенсона.

Для модулярного експоненіювання $a^x mod\ p$ (вважаємо, що $x \le \varphi(p)$, $\varphi(p)$ – значеня функції Ейлера від модуля p) використаємо проміжну матрицю, представлену в табл. 9.17. Її розмірність дорівнює розрядності n модуля p. В стовбцях матриці записано величини a^{2^i} mod p в базисі Радемахера, тобто a_{ij}=0, 1. Тоді будь–який степінь x можна записати за степенями 2 і шуканий результат можна отримати, перемноживши відповідну кількість стовбців за допомогою табл. 9.16. Основними перевагами такого методу є здійснення операцій в системі залишкових класів, а не оперувати з великими числами, що дозволяє пришвидшити алгоритм модулярного експоненціювання.

Таблиця 9.17 – МАТРИЦЯ ПІДНЕСЕННЯ ДО СТЕПЕНЯ В БАЗИСІ РАДЕМАХЕРА–КРЕСТЕНСОНА

$a_{n-1\ n-1}$	…	$a_{i\ n-1}$		$a_{1\ n-1}$	$a_{0\ n-1}$
…	…	…	…	…	…
$a_{n-1\ j}$	…	$a_{i\ j}$	…	$a_{1\ j}$	$a_{0\ j}$
…	…	…	…	…	…
$a_{n-1\ 1}$	…	$a_{i\ 1}$	…	$a_{1\ 1}$	$a_{0\ 1}$
$a_{n-1\ 0}$	…	$a_{i\ 0}$	…	$a_{1\ 0}$	$a_{0\ 0}$
$a^{2^{n-1}}$	…	a^{2^i}	…	a^{2^1}	a^{2^0}

Отже, для модулярного експоненіювання $a^x(modp)$ потрібно скористатися алгоритмом піднесення до степеня двійкового числа будь-якої розрядності за модулем *p*.

Пропонований алгоритм піднесення до степеня двійкового числа будь-якої розрядності за модулем *p* дозволяє зменшити часову складність за рахунок заміни операції множення операцією додавання, підвищити швидкодію на 30-40% (рис.9.5), для чисел розрядності менше 256 біт, а в діапазоні від 256 біт на

10%. Чисельний експеримент показав, що запропонований алгоритм піднесення до степеня двійкового числа будь-якої розрядності за модулем p в базисі Радемахера–Крестенсона дозволяє зменшити складність з $O3 = \frac{n^3}{2}$ до

$$O4(n) = \begin{cases} \log_2 n \cdot \left(\frac{n}{2} \cdot \log_2 n + 1 \right), \textit{якщо } n < 256 \\ n \cdot \left(\frac{n}{2} \cdot \log_2 n + 1 \right), \textit{в інших випадках.} \end{cases}$$

в

$$E2(n) = \begin{cases} \frac{n}{(\log_2 n)^2 + 2 \cdot \log_2 n}, \textit{для } n < 256 \\ \frac{n}{(\log_2 n) + 2 \cdot \log_2 n}, 256 \le n \le 1024 \end{cases}$$

разів, що показано на рис. 9.6.

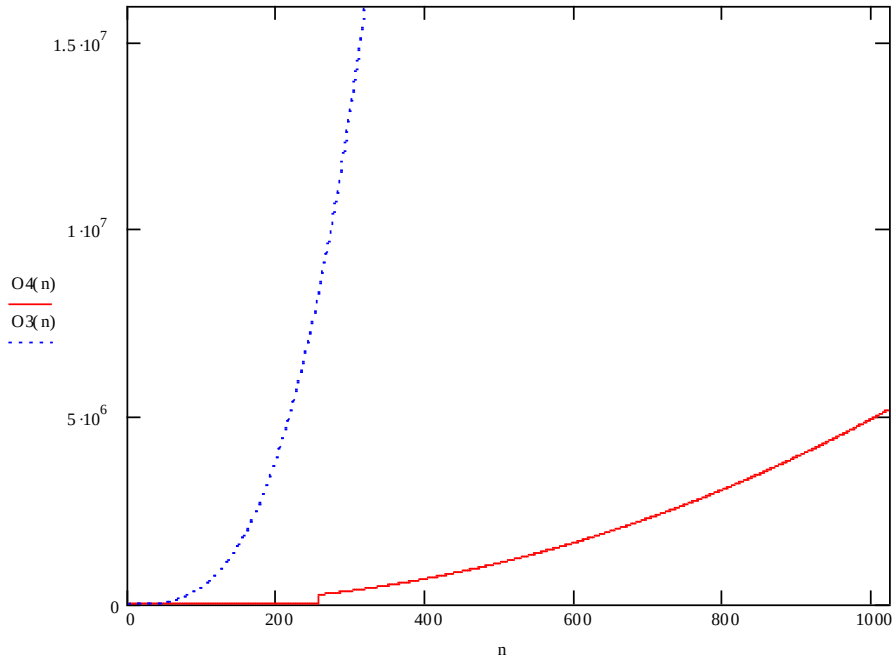


Рисунок 9.5 – Часова складність операції модулярного піднесення до степеня.

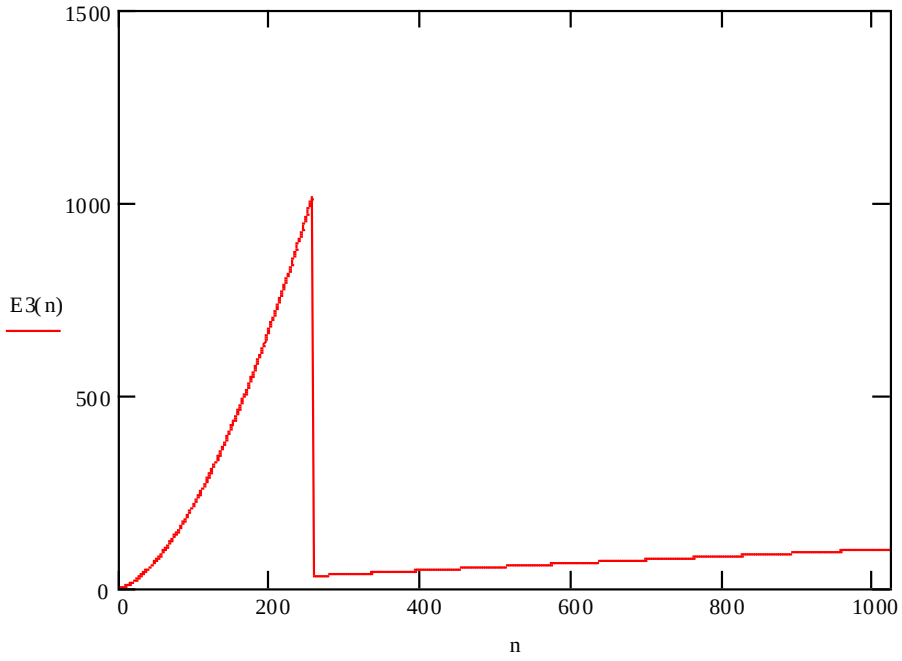


Рисунок 9.6 – Ефективність запропонованого алгоритму.

Дослідження показали, що алгоритм запропонований алгоритм характеризується високою швидкодією та ефективністю для знаходження значення операції піднесення до степеня двійкового числа будь-якої розрядності за модулем *p*. Злід зазначити, що при збільшенні розрядності чисел зменшується ефективність (рис 9.6), бо частина ресурсу комп'ютера буде задіяна на розв'язок службової інформації, що значно збільшує складність і кількість операцій та зменшує швидкодію. Оскільки, операція модулярного експоненіювання є базовою в найбільш поширених системах захисту інформаційних потоків з відкритими ключами (RSA, Ель–Гамаля тощо), визначення стійкості еліптичних кривих методом пошуку їх порядку за допомогою алгоритму Шуфа, то доцільно використовувати розроблений метод в задачах захисту інформаційних потоків на практиці для вдосконалення систем захисту інформаційних потоків.

Лекція 12

Теорія алгоритмів RSA та Ель-Гамаля в розмежованій системі числення Радемахера-Крестенсона

Аналіз наукових тенденцій розвитку теорії та перспективних інформаційних технологій покращення ефективності опрацювання інформаційних потоків в комп'ютерних мережах, проведений на основі новітніх публікацій, потребує поглибленого дослідження теоретичних засад базисів Крестенсона та Радемахера. Слід зауважити, що найбільш фундаментально досліджено цілочисельну форму в системі залишкових класів, яка утворюється на основі прямого перетворення ТЧБ Крестенсона.

Тому є доцільним дослідити інші форми систем залишкових класів, які можуть бути використані для реалізації високопродуктивних алгоритмів опрацювання і захисту інформаційних потоків, а також виконати порівняльний аналіз різних ТЧБ з базисом Радемахера, який породжує двійкову систему числення на основі відповідних критеріїв, та дослідити часову складність алгоритмів шифрування інформаційних потоків з використання алгоритмів RSA, Ель-Гамаля в розмежованій системі числення Радемахера-Крестенсона.

Незважаючи на ефективні алгоритми формування структуризованих даних, які реалізуються на основі програмно-апаратних мультибазисних процесорів і забезпечують захист від помилок та певний рівень захисту інформації (ЗІ) від несанкціонованого доступу, який не відповідає умовам сучасного рівня ЗІ. Тому потрібно додаткове опрацювання інформаційних потоків, формування захищених даних, що потребує аналізу і ефективності існуючих методів захисту структуризованих даних на основі відомих алгоритмів RSA, Ель-Гамаля.

Оцінка часових складностей алгоритмів опрацювання інформації в задачах криптографії.

Сучасний розвиток комп'ютерної техніки потребує високого рівня захисту інформації, додаткового опрацювання інформаційних потоків та формування захищених даних. Тому зроблений аналіз ефективності існуючих методів захисту структуризованих даних на основі відомих алгоритмів RSA, Ель-Гамаля, а особливо алгоритмів з використанням математичного апарату

еліптичних кривих, говорить, що їх використання є перспективним для вдосконалення захисту інформаційних потоків.

Система захисту інформаційних потоків від несанкціонованого доступу з відкритим ключем RSA базується на задачі множення і розкладу чисел на прості множники, які є часово однонаправленими задачами. В системі RSA кожний з учасників процесу шифрування має в розпорядженні відкритий і закритий (секретний) ключі, кожний з яких складається з пари цілих чисел, які генеруються наступним чином [4]:

1. Генеруються два випадкових простих числа p і q довільного розміру (чим більші, тим краще для стійкості системи захисту інформаційних потоків).

2. Обчислюється $n = p * q$, тобто модуль криптоперетворень з використанням матричних перетворень: подаємо число p і q у вигляді: $p = p_{r-1}2^{r-1} + p_{r-2}2^{r-2} + \ldots + p_1 2^1 + p_0 2^0$, $q = q_{r-1}2^{r-1} + q_{r-2}2^{r-2} + \ldots + q_1 2^1 + q_0 2^0$, де r-розрядність чисел p і q. Для знаходження результату їх множення побудуємо матрицю, представлену в табл. 9.18, де $m_{ij} = 2^{i+j}$.

Добуток чисел p і q отримуємо за формулою:

$$n = p \cdot q = \sum_{s,k=1}^{r-1} m_{sk} \qquad (9.11)$$

де $p_s, q_k = 1$, тобто m_{sk} знаходиться на перетині стовбця та рядка, для яких відповідні p_i і q_j дорівнюють 1. Це значно зменшить складність алгоритму пошуку модуля криптоперетворень $O\left(3{,}5n^2 + n - \frac{1}{2}\right)$.

Таблиця 9.18 – Матриця знаходження модуля перетворення в базисі Радемахера–Крестенсона

	q_{r-1}	…	q_j	…	q_1	q_0
p_{r-1}	$m_{r-1\ r-1}$	…	$m_{r-1\ j}$	…	$m_{r-1\ 1}$	$m_{r-1\ 0}$

…	…	…	…	…	…	…
p_i	$m_{i\ r-1}$	…	m_{ij}	…	m_{i1}	m_{i0}
…	…	…	…	…	…	…
p_1	$m_{1\ r-1}$	…	m_{1j}	…	m_{11}	m_{10}
p_0	$m_{0\ r-1}$	…	m_{0j}	…	m_{01}	m_{00}

3. Аналогічним чином знаходимо значення функції Ейлера від числа n, а саме $\varphi(n)=(p-1)(q-1)$.

4. Вибираємо ціле число $e, 1<e<\varphi(n)$ - взаємнопросте з $\varphi(n)$. Його доцільно вибирати з найменшою кількістю одиничних бітів в двійковій формі, рекомендуються вибирати числа Ферма 17, 257, 65537…Число e називають відкритою експонентою; час шифрування залежить від швидкодії операції піднесення числа в степінь по модулю, тому малі значення e можуть зменшити стійкість системи захисту інформаційних потоків.

5. Знаходимо секретну експоненту d обернену до числа e за модулем $\varphi(n)$, тобто $d\cdot e\equiv 1(mod\ \varphi(n))$. В класичній літературі знаходження оберненого елемента в залишкових классах обчислюється згідно розширеного алгоритму Евкліда. Для зменшення складності доцільно скористатися матричним методом:

Розглянемо модуль $\varphi(n)$. Нехай x пробігає зведену систему найменших додатних лишків за модулем $\varphi(n)$: $r_1=1, r_2=2, r_3=3,\ldots, r_i=i,\ldots, r_{\varphi(n)-1}=\varphi(n)-1$.

Тоді для числа $e<\varphi(n)$ добуток $e\cdot x$ теж пробігатиме зведену систему лишків за цим модулем:

$$\begin{cases} e\cdot r_1 \bmod \varphi(n)=c_1 \\ e\cdot r_2 \bmod \varphi(n)=c_2 \\ e\cdot r_3 \bmod \varphi(n)=c_3 \\ \ldots\ldots\ldots\ldots\ldots \\ e\cdot r_i \bmod \varphi(n)=c_i \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ e\cdot r_{\varphi(n)-1} \bmod \varphi(n)=c_{\varphi(n)-1}. \end{cases} . \qquad (9.12)$$

Аналогічно для числа d:

$$\begin{cases} d \cdot r_1 \bmod \varphi(n) = g_1 \\ d \cdot r_2 \bmod \varphi(n) = g_2 \\ d \cdot r_3 \bmod \varphi(n) = g_3 \\ \ldots\ldots\ldots\ldots\ldots \\ d \cdot r_i \bmod \varphi(n) = g_i \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ d \cdot r_{\varphi(n)-1} \bmod \varphi(n) = g_{\varphi(n)-1}. \end{cases} . \quad (9.13)$$

З першої системи виберемо перше рівняння, а з другого – рівняння, в якому $c_1 = r_i$: $e \cdot r_1 \, mod \, \varphi(n) = c_1$

$$d \cdot r_i \, mod \, \varphi(n) = g_i . \quad (9.14)$$

Перемножимо ці рівняння:

$$er_1 \cdot dr_i \, mod \, \varphi(n) = c_1 \cdot g_i . \quad (9.15)$$

Враховуючи, що $r_1 = 1, c_1 = r_i$, отримаємо добуток за відповідним модулем:

$$e \cdot d \, mod \, \varphi(n) = g_i . \quad (9.16)$$

Даним методом можна перемножувати будь–яку кількість множників, підносити до будь–якого степеня, шукати обернені елементи та квадратні корені. Для цього зручно побудувати таблицю Келі (табл. 9.19) наприклад, для $\varphi(n)$=20, для $p = 3, q = 11$.

6. В результаті отримуємо пари $P = (e,n)$ - відкритий ключ, $S = (d,n)$ - таємний ключ.

Після генерування ключів, шифрування інформаційних потоків здійснюється наступним чином:

- Беремо $P = (e,n)$ і інформаційний потік у вигляді цілого числа $M \in D \in Z_n, 0 < M < n-1$.

Таблиця 9.19 – Таблиця Келі

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	**4**	6	8	10	12	14	16	18	20	22	1	3	5	7	9	11	13	15
3	6	**9**	12	15	18	21	1	4	7	10	13	16	19	22	2	5	8	11
4	8	12	**16**	20	1	5	9	13	17	21	2	6	10	14	18	22	3	7
5	10	15	20	**2**	7	12	17	22	4	9	14	19	1	6	11	16	21	3
6	12	18	1	7	**13**	19	2	8	14	20	3	9	15	21	4	10	16	22
7	14	21	5	12	19	**3**	10	17	1	8	15	22	6	13	20	4	11	18
8	16	1	9	17	2	10	**18**	3	11	19	4	12	20	5	13	21	6	14
9	18	4	13	22	8	17	3	**12**	21	7	16	2	11	20	6	15	1	10
10	20	7	17	4	14	1	11	21	**8**	18	5	15	2	12	22	9	19	6
11	22	10	21	9	20	8	19	7	18	**6**	17	5	16	4	15	3	14	2
12	1	13	2	14	3	15	4	16	5	17	**6**	18	7	19	8	20	9	21
13	3	16	6	19	9	22	12	2	15	5	18	**8**	21	11	1	14	4	17
14	5	19	10	1	15	6	20	11	2	16	7	21	**12**	3	17	8	22	13
15	7	22	14	6	21	13	5	20	12	4	19	11	3	**18**	10	2	17	9
16	9	2	18	11	4	20	13	6	22	15	8	1	17	10	**3**	19	12	5
17	11	5	22	16	10	4	21	15	9	3	20	14	8	2	19	**13**	7	1
18	13	8	3	21	16	11	6	1	19	14	9	4	22	17	12	7	**2**	20
19	15	11	7	3	22	18	14	10	6	2	21	17	13	9	5	1	20	**16**

- Шифрується інформаційний потік, і передається по каналу зв'язку згідно співвідношення $P(M) = M^e \bmod n$, використовуючи алгоритм піднесення до степеня двійкового числа будь-якої розрядності за модулем *p*. Її розмірність дорівнює розрядності n модуля p. В стовбцях матриці записано величини $M^{2^i} (\bmod n)$ в базисі Радемахера, тобто M_{ij}=0, 1 (табл.9.20). Тоді будь–який степінь e можна записати за степенями 2 і шуканий результат можна отримати, перемноживши відповідну кількість стовбців за допомогою табл. 9.20.

Таблиця 9.20 – Матриця шифротексту алгоритму шифрування RSA в базисі Радемахера–Крестенсона

$M_{n-1\ n-1}$	…	$M_{i\ n-1}$		$M_{1\ n-1}$	$M_{0\ n-1}$
…	…	…	…	…	…
$M_{n-1\ j}$	…	$M_{i\ j}$	…	$M_{1\ j}$	$M_{0\ j}$

…	…	…	…	…	…
$M_{n-1\,1}$	…	$M_{i\,1}$	…	$M_{1\,1}$	$M_{0\,1}$
$M_{n-1\,0}$	…	$M_{i\,0}$	…	$M_{1\,0}$	$M_{0\,0}$
$M^{2^{n-1}}$	…	M^{2^i}	…	M^{2^1}	M^{2^0}

Для дешифрування використовується таємний ключ $S=(d,n)$ до зашифрованного інформаційного потоку $P(M)$: $S(P(M))=(P(M))^d \bmod n$.

Використання ТЧБ Радемахера в алгоритмі шифрування інформаційних потоків RSA дозволяє зменшити часову складність, яка базується на складності виконання операції $P(M)=M^e \bmod n$. Як показано в літературі, для виконання цієї операції з використанням алгоритму швидкого піднесення до степеня потрібно $O(\ln e)$ операцій множень по модулю. Отже, з врахуванням того, що операція модулярного множення має складність $O(r^2)$, то складність $P(M)=M^e \bmod n$ буде $O(n^2 \cdot \ln e)$, де n - розрядність модуля n. Оскільки в алгоритмі RSA для шифрування та дешифрування використовується одна і та ж операції, то складність цього алгоритму оцінюється, як $O1(n^2 \cdot \ln e + 2n)$, тоді як запропонованого алгоритму з використанням ТЧБ Радемахера

$$O3(n)=\begin{cases} \log_2 n\left(3\log_2 n+\dfrac{n}{2}\right), якщо\ n<256 \\ n\cdot\left(3\log_2 n+\dfrac{n}{2}\right), в\ інших\ випадках \end{cases}$$ (рис. 9.7).

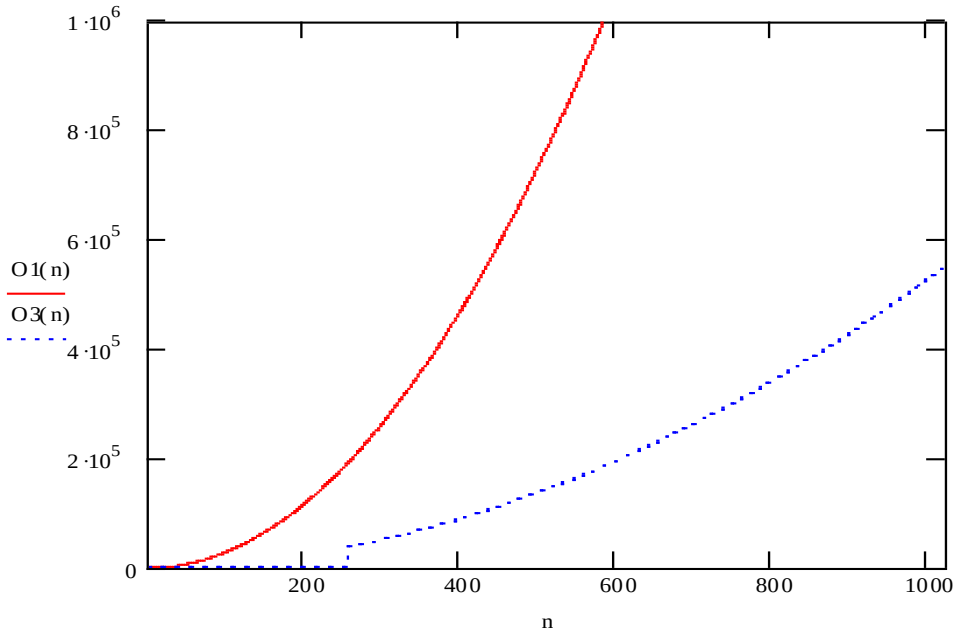


Рисунок 9.7 – Часові складності $O3(n)$ - алгоритму шифруваня RSA з використанням розмежованої системи числення Радемахера-Крестенсона, $O1(n)$ - класичного алгоритму RSA.

Основною трудомісткою операцією при шифруванні та дешифруванні інформаційних потоків в алгоритмі RSA є операція модулярного множення та експоненціювання, тому використання запропонованих алгоритмів дозволить значно зменшити складності на 20-40% для параметрів, менших 256 біт, і на 10 % при параметрах від 256 до 1024 біт.

Для реалізації алгоритму шифрування Ель-Гамаля потрібно першочергово згенерувати ключі згідно наступної послідовності дій:

1. Генерується випадкове просте число p довжини n.
2. Вибирається довільне ціле число g, яке є первісним коренем по модулю p, та будь-яке число $x \in (1, p)$ взаємно просте з $p-1$.

Обчислюємо відкритий ключ $y = g^x mod\ p$ з використанням матричного методу піднесення до степеня в базисі Радемахера–Крестенсона. Записуємо в стовпцях матриці величини $g^{2^i}(mod\, n)$ в базисі Радемахера, тобто g_{ij}=0, 1 (табл.9.21) та подамо степінь x за степенями 2, тобто $x = x_{n-1}2^{n-1} + \ldots + + x_i 2^i + \ldots + x_1 2 + x_0$. Шуканий результат можна отримати, перемноживши відповідну кількість стовбців за допомогою табл. 9.17. Отже, матричним методом

експоненціювання можна знайти відкритий ключ алгоритму шифрування Ель-Гамаля.

Таблиця 9.21 – МАТРИЦЯ ЗНАХОДЖЕННЯ ВІДКРИТОГО КЛЮЧА АЛГОРИТМУ ШИФРУВАННЯ ЕЛЬ -ГАМАЛЯ В БАЗИСІ РАДЕМАХЕРА–КРЕСТЕНСОНА

$g_{n-1\ n-1}$	…	$g_{i\ n-1}$		$g_{1\ n-1}$	$g_{0\ n-1}$
…	…	…	…	…	…
$g_{n-1\ j}$	…	$g_{i\ j}$	…	$g_{1\ j}$	$g_{0\ j}$
…	…	…	…	…	…
$g_{n-1\ 1}$	…	$g_{i\ 1}$	…	$g_{1\ 1}$	$g_{0\ 1}$
$g_{n-1\ 0}$	…	$g_{i\ 0}$	…	$g_{1\ 0}$	$g_{0\ 0}$
$g^{2^{n-1}}$	…	$g^{2^{i}}$	…	$g^{2^{1}}$	$g^{2^{0}}$

Відкритим ключем в алгоритмі шифрування інформаційних потоків Ель-Гамаля є трійка *(p,g,y)*, закритим – число *x*. Після генерування ключів, шифрування повідомлення M здійснюється згідно алгоритму:

1. Вибирається випадково секретне число k, взаємно просте з p − 1.
2. Обчислюється $a = g^k mod\ p$, $b = y^k M mod\ p$, де M — інформаційний потік який шифрується. Для обчислення $b = y^k M mod\ p$ потрібно послідовно застосувати алгоритм модулярного експоненціювання та модулярного множення в розмежованій системі числення Радемахера – Крестенсона.

Пара чисел *(a,b)* називається шифротекстом, причому довжина шифротексту в алгоритмі шифрування інформаційних потоків Ель-Гамаля вдвоє більша від повідомлення M.

Таким чином, знаючи секретний ключ *x*, вихідне повідомлення можна обчислити з шифротексту *(a,b)* по формулі: $M = b \cdot (a^x)^{-1} mod\ p$.

Стійкість даної системи базується на часовій складності задачі дискретного логарифмування. Як і в алгоритмі шифрування RSA, так і в Ель-Гамаля основною операцією є модулярне експоненціювання. Тому часова складність алгоритму Ель-Гамаля, з врахуванням генерування ключів, буде $O(n^2(3n+1))$. Використання розмежованої системи числення Радемахера –

Крестенсона дозволяє зменшити складність з $O1(n^2(3n+1))$ до

$$O3(n)=\begin{cases}\log_2 n\cdot\left(\log_2 n+\frac{n}{2}\log_2 n+1\right), якщо\ n<256\\ n\cdot\left(\log_2 n+\frac{n}{2}\log_2 n+1\right), 256\le n\le 1024\end{cases},$$ де n- розрядність модуля p (рис.9.8).

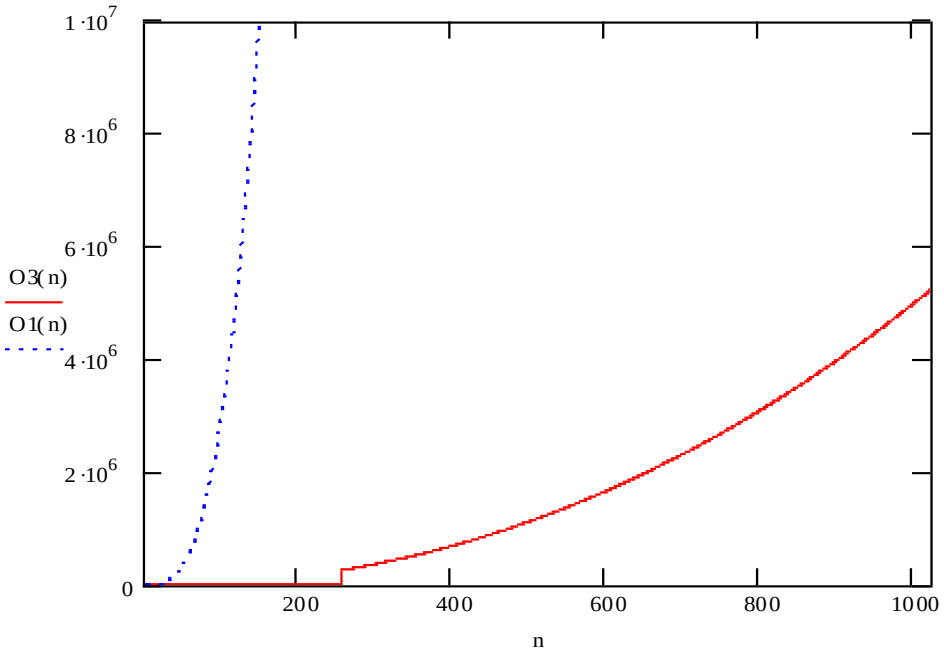


Рисунок 9.8 – Складності $O3(n)$ - алгоритму шифруваня Ель-Гамаля з використанням розмежованої системи числення Рademахера-Крестенсона, $O1(n)$ - класичного алгоритму Ель-Гамаля.

Отже, використання розмежованої системи числення Рademахера – Крестенсона в задачах захисту інформаційних потоків на основі алгоритмів RSA та Ель-Гамаля, дозволяє ефективно застосовувати матричні методи при побудові мультибазисних процесорів шифрування, а заміна операцій множень операціями додавання на етапах генерування ключів, шифрування та дешифрування – дозволяє значно зменшити часову складність від 10 до 40 % в залежності від розрядності параметрів алгоритмів RSA та Ель-Гамаля.