МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського»
Фізико-технічний інститут

Методичні вказівки до виконання розрахункової роботи з кредитного модуля «Симетрична криптографія»

(проект)

для студентів спеціальності 113 Прикладна математика

(Колись буде) Рекомендовано Вченою радою Фізико-технічного інституту

Київ
КПІ ім. Ігоря Сікорського
2019

Методичні вказівки до виконання розрахункової роботи з кредитного модуля «Симетрична криптографія» / Уклад. М.М. Савчук, Л.А. Завадська, С.В. Яковлєв. – К.: НТУУ «КПІ», 2019. – 21 с., 4 джерела.

Гриф надано Вченою радою Фізико-технічного інституту КПІ ім. Ігоря Сікорського (Протокол № від)

Навчальне видання

Методичні вказівки до виконання розрахункової роботи з кредитного модуля «Симетрична криптографія»

для студентів спеціальностей 113 Прикладна математика

Укладачі: Савчук Михайло Миколайович, д.ф.-м.н., доцент
Завадська Людмила Олексіївна, к.ф.-м.н., доцент
Яковлєв Сергій Володимирович, к.т.н., доцент

Відповідальний редактор: М.М. Савчук, д.ф.-м.н., доцент

Рецензент: *наприклад*, О.Є. Архипов, д.т.н., професор

ЗМІСТ

Мета та задачі розрахункової роботи

З метою кращого засвоєння матеріалу курсу та набування навичок самостійної роботи студентам пропонується виконати розрахункову роботу (РР) за розділом «Булеві функції та їх криптографічні властивості». Виконання завдань РР сприяє більш поглибленому вивченню студентами теоретичного матеріалу, формуванню вмінь використовувати знання для розв'язання практичних задач. При виконанні РР студент має продемонструвати володіння методами представлення булевих функцій у табличному вигляді, нормальних формах та спектральному розкладі, знання методів оцінювання криптографічних параметрів булевих функцій, навички програмування.

Розрахункова робота виконується студентом САМОСТІЙНО із забезпеченням необхідних консультацій з окремих питань з боку викладачів.

Наявність позитивної оцінки, отриманої студентом за виконання розрахункової роботи, є необхідною умовою допуску до семестрового контролю з дисципліни.

Короткі теоретичні відомості

1 Визначення булевих функцій

Одновимірна булева функція – це перетворення двійкового вектора в одиничний біт, яке можна розглядати як відображення $f:\{0,1\}^n \to \{0,1\}$ або, якщо позначити через $V_n = \{0,1\}^n$ множину усіх двійкових векторів довжини *n*, як відображення $f:V_n \to \{0,1\}$.

Багатовимірна булева функція – це перетворення двійкового вектора у двійковий вектор, яке можна розглядати як відображення $F:V_n \to V_m$, де *n* та *m* – розмірності вхідного та вихідного векторів.

Там, де потрібно точно вказати розмірності входів та виходів, ми будемо писати $n-f$ для одновимірної та $(n,m)-F$ для багатовимірної булевої функції.

Кожна багатовимірна булева функція може бути представлена у вигляді вектора одновимірних *координатних функцій*, що повертають лише певний біт вихідного вектора: $F=(f_1, f_2, ..., f_m)$.

Існує декілька форм представлення булевих функцій, що є зручними для тих чи інших застосувань. Розглянемо найважливіші серед них.

1.1 Таблиця істинності

Найпростіший спосіб представлення булевої функції полягає у переліку всіх можливих вхідних значень та визначення відповідних їм значень виходу. Такий перелік зазвичай оформлюється у вигляді таблиці, яку називають *таблицею істинності*. Загальний вид таблиці істинності наведено на рис. 1.

Такий спосіб представлення є вичерпним та наглядним, але він вимагає великого об'єму пам'яті та тому незручний у практичному використанні для великих значень *n* та *m*. Втім, коли таблицю істинності можливо застосувати, вона надає найшвидший спосіб роботи із булевою функцією.

	x_1	x_2	...	x_{n-1}	x_n	f_1	f_2	...	f_m
2^n штук	0	0	...	0	0	$\alpha_1^{(1)}$	$\alpha_2^{(1)}$	…	$\alpha_m^{(1)}$
	0	0	...	0	1	$\alpha_1^{(2)}$	$\alpha_2^{(2)}$	…	$\alpha_m^{(2)}$
	⋮	⋮		⋮	⋮	⋮	⋮		⋮
	1	1	…	1	0	$\alpha_1^{(2^n-1)}$	$\alpha_2^{(2^n-1)}$	…	$\alpha_m^{(2^n-1)}$
	1	1	…	1	1	$\alpha_1^{(2^n)}$	$\alpha_2^{(2^n)}$	…	$\alpha_m^{(2^n)}$

Рисунок 1 – Таблиця істинності для $(n,m)-F$. Тут $\alpha_i^{(j)}$ – довільні бітові значення.

1.2 Алгебраїчна нормальна форма

Поліномом Жегалкіна від n змінних називають канонічний поліном над $GF(2)$ вигляду:

$$P(x_1,...,x_n) = a_0 \oplus \bigoplus_{k=1}^{n} \bigoplus_{1\le i_1<...<i_k\le n} a_{i_1,...i_k} x_{i_1}...x_{i_k}$$

де всі коефіцієнти $a_* \in \{0,1\}$.

Будь-яка функція $n-f$ може бути єдиним чином зображена у вигляді полінома Жегалкіна. Таке представлення називається *алгебраїчною нормальною формою* булевої функції.

Алгебраїчний степінь $\deg(f)$ булевої функції $n-f$ – це максимальний степінь доданків із ненульовими коефіцієнтами у зображенні функції поліномом Жегалкіна. Очевидно, що максимальний степінь функції $n-f$ не може перевищувати *n*.

Алгебраїчний степінь $\deg(F)$ булевої функції $(n,m)-F$ – це максимальний алгебраїчний степінь її координатних функцій.

Пошук алгебраїчної нормальної форми можна виконати за таким алгоритмом (що є модифікацією так званого *перетворення Мебіуса*). Нехай e_i – вектор, в якому *i*-тий біт дорівнює 1, а всі інші – 0. Тоді, як неважко помітити, виконуються такі співвідношення:

$$a_0 = f(0,0,0,...,0),$$
$$a_i = f(e_i) \oplus a_0,$$
$$a_{ij} = f(e_i \oplus e_j) \oplus a_i \oplus a_j \oplus a_0,$$
$$a_{ijk} = f(e_i \oplus e_j \oplus e_k) \oplus a_{ij} \oplus a_{jk} \oplus a_{ik} \oplus a_i \oplus a_j \oplus a_k \oplus a_0$$

і так далі, тобто коефіцієнти поліному обчислюються послідовно, від молодших степенів до старших.

Пошук АНФ функції $(n,m)-F$ виконується таким самим чином, при цьому одразу обчислюються коефіцієнти $a_* \in \{0,1\}^n$, в яких кожна координата є коефіцієнтом АНФ відповідної координатної функції. Швидкі методи обчислення АНФ наведені у розділі 5.

1.3 Розклад у ряд Фур'є

Сімейство функцій $g_a(x) = (-1)^{a \cdot x}$, $\forall a \in V_n$, де символом $a \cdot x$ позначено скалярний добуток двох бітових векторів, є ортогональним, тобто виконується така рівність:

$$\sum_{x \in V_n} (-1)^{a \cdot x} (-1)^{b \cdot x} = \begin{cases} 0, & a \neq b \\ 2^n, & a = b \end{cases}.$$

Тому кожна булева функція $n - f$ може бути представлена у вигляді *ряду Фур'є*:

$$f(x) = \sum_{a \in V_n} c_f(a) \cdot (-1)^{a \cdot x}, \text{ де } c_f(a) = \frac{1}{2^n} \sum_{x \in V_n} f(x)(-1)^{a \cdot x}.$$

Коефіцієнти $c_f(a)$ називають *коефіцієнтами Фур'є* булевої функції *f*, а множину $\{c_f(a), a \in V_n\}$ – *спектром Фур'є*. Розклад у ряд Фур'є є однозначним, тому спектр повністю описує функцію *f*.

1.4 Перетворення Уолша-Адамара

Перетворенням Уолша-Адамара булевої функції $n - f$ називають перетворення такого виду:

$$W_f(a) = \sum_{x \in V_n} (-1)^{f(x) \oplus a \cdot x}.$$

У деяких джерелах це перетворення називають просто *перетворенням Уолша*.

Цілочисельні коефіцієнти $W_f(a)$ називають *коефіцієнтами Уолша-Адамара* (або просто *коефіцієнтами Уолша*) булевої функції *f*, а їх множину – *спектром Уолша*. Для коефіцієнтів Уолша-Адамара виконується нерівність:

$$\min_a \left| W_f(a) \right| \leq 2^{n/2} \leq \max_a \left| W_f(a) \right|,$$

яка випливає з так званої *рівності Парсеваля*: $\sum_a W_f^2(a) = 2^{2n}$.

Множина коефіцієнтів Уолша-Адамара однозначно описує вихідну булеву функцію *f*. Зокрема, можна відновити *f* через обернене перетворення Уолша-Адамара:

$$(-1)^{f(x)} = \frac{1}{2^n} \sum_a W_f(a)(-1)^{a \cdot x}.$$

Між коефіцієнтами Фур'є та коефіцієнтами Уолша-Адамара виконується співвідношення: $W_f(a) = 2^n \cdot \delta_{a,0} - 2^{n+1} c_f(a)$ (тут $\delta_{x,y}$ – символ Кронекера). Тому іноді перетворення Уолша-Адамара називають *дискретним перетворенням Фур'є*.

2 Криптографічні властивості булевих функцій

Для застосування у криптографічних задачах булеві функції повинні мати додаткові властивості, що унеможливлюють або значно ускладнюють криптоаналіз. Розглянемо найважливіші з таких властивостей.

2.1 Невиродженість

Змінна x_k називається *істотною* для булевої функції $(n,m)-F$, якщо існує такий набір значень інших змінних (a_i), що:

$$F(a_1,...,a_{k-1},0,a_{k+1},...,a_n) \neq F(a_1,...,a_{k-1},1,a_{k+1},...,a_n)$$

Булева функція $(n,m)-F$ називається *невиродженою*, якщо всі її змінні істотні.

Іноді розглядають більш суворе формулювання невиродженості: булева функція називається невиродженою тоді, коли всі її змінні істотні для кожної координатної функції. В цьому випадку гарантується, що кожен біт входу впливає на кожен біт виходу, а тому аналітик не може знизити складність функції за рахунок зменшення кількості вхідних змінних. Звісно, для одномірних булевих функцій ці визначення співпадають.

Невиродженість гарантує, що кожен біт входу впливає на кожен біт виходу, а тому аналітик не може знизити складність функції за рахунок зменшення кількості вхідних змінних.

Існує простий спосіб перевірки змінної на істотність: неістотні змінні не входять у алгебраїчну нормальну форму булевої функції. Відповідно, функція є невиродженою, якщо її АНФ включає усі вхідні змінні.

2.2 Збалансованість та k-збалансованість

Булева функція $(n,m)-F$ називається *збалансованою* (або *рівноймовірною*), коли кожне значення функції досягається на рівній кількості значень входу, тобто:

$$\forall y \in V_m: \quad \#\{x \in V_n : F(x) = y\} = 2^{n-m}.$$

Для одновимірних функцій поряд із означенням збалансованості розглядають також поняття дисбалансу. *Дисбалансом* булевої функції $n-f$ називається величина

$$disbal(f) = \left|\#\{x \in V_n : f(x) = 0\} - \#\{x \in V_n : f(x) = 1\}\right|.$$

Дисбаланс показує, наскільки імовірність одержати одне значення на виході булевої функції більше за імовірність одержати інше. У збалансованих функцій дисбаланс дорівнює нулю. Також легко показати, що дисбаланс простим шляхом визначається через перетворення Уолша-Адамара: $disbal(f) = \left|W_f(0)\right|$.

Булева функція називається *k-збалансованою* (або *k-рівноймовірною*), якщо кожна її $(n-k)$-підфункція, одержана фіксацією довільних *k* вхідних змінних довільними значеннями, буде рівноймовірною.

Встановлено, що функція $n-f$ буде *k*-збалансованою тоді та тільки тоді, коли її коефіцієнти Уолша задовольняють умові: $\forall a, 1 \le wt(a) \le k : W_f(a) = 0$.

Збалансованість булевої функції є важливою характеристикою: якщо функція збалансована, аналітик за значеннями виходу не одержує додаткової інформації про характер даних на вході. Однак не всі збалансовані функції є однаково гарними.

Двоїста функція для заданої булевої функції $f(x_1, x_2, ..., x_n)$ визначається як $f^*(x_1, x_2, ..., x_n) = f(x_1 \oplus 1, x_2 \oplus 1, ..., x_n \oplus 1) \oplus 1$, тобто інверсією всіх входів та виходу. Функція називається *самодвоїстою*, якщо $f(x_1, x_2, ..., x_n) = f^*(x_1, x_2, ..., x_n)$. Звісно, що самодвоїсті функції є збалансованими; однак завдяки легкості маніпулювання вхідними даними вони не є криптографічно стійкими.

2.3 Кореляційний імунітет

Нехай на множині вхідних векторів x (тобто, на множині V_n) задано рівномірний імовірнісний розподіл. Тоді для даної булевої функції $(n,m)-F$ індукується імовірнісний розподіл і на множині значень $F(x)$ (тобто, на множині V_m).

Булева функція $(n,m)-F$ має *кореляційний імунітет k-того порядку*, якщо для довільного набору індексів $1 \le i_1 < ... < i_k \le n$ та довільного біту виходу $1 \le r \le m$ взаємна інформація між вектором вхідних змінних з обраними індексами та відповідним бітом виходу дорівнює нулю:

$$I\left(\left(x_{i_1}, ..., x_{i_k}\right), y_r\right) = 0,$$

де взаємна інформація визначається звичайним шляхом через ентропію:

$$I(X,Y) = H(X) + H(Y) - H(X,Y).$$

Підкреслимо, що говорити про кореляційний імунітет можна лише тоді, коли на множині значень входу заданий імовірнісний розподіл. Більш того, оскільки вхідні значення розподілені рівноімовірно, то означення кореляційного імунітету *k*-того порядку співпадає з означенням *k*-збалансованості для всіх координатних функцій, що не є тотожними константами. Відповідно, функція матиме кореляційний імунітет порядку *k* тоді та тільки тоді, коли її коефіцієнти Уолша задовольняють умові: $\forall a, 1 \le wt(a) \le k : W_f(a) = 0$.

Також встановлено, що якщо функція $n-f$ має кореляційний імунітет *k*-того порядку, то її алгебраїчний степінь не перевищує $n-k$; якщо додатково $k \le n-2$ і функція є збалансованою, то її алгебраїчний степінь не вище за $n-k-1$ (нерівність Зігенталера).

Кореляційний імунітет дозволяє уникати так званих *кореляційних атак*, коли аналітик, використовуючи кореляційні залежності між бітами входу та виходу, може відновити окремі біти входу (зазвичай, біти ключа).

2.4 Нелінійність

Лінійною булевою функцією називається функція, яку можна представити у вигляді $l_a(x) = a \cdot x$, де $a \in V_n$.

Афінною булевою функцією називається функція, яку можна представити у вигляді $b_a(x) = a \cdot x \oplus c$, де $a \in V_n$, $c \in \{0,1\}$. Множину всіх афінних функцій від *n* змінних позначимо через A_n.

Лінійні та афінні функції з точки зору криптографії вважаються слабкими, оскільки вони можуть бути швидко та ефективно обчислені та обернені за допомогою апарату лінійної алгебри та чисельних методів. Функції, що використовуються у криптографії, повинні мати якомога більшу нелінійність.

Нелінійністю булевої функції $n-f$ називається її відстань до множини A_n в термінах відстані Хемінга:

$$NL_f = \min_{l\in A_n}(dist(f,l)) = \min_{l\in A_n}(wt(f \oplus l)) = \min_{l\in A_n}\left(\sum_x f(x)\oplus l(x)\right),$$

де остання сума ведеться в арифметичному сенсі.

Для нелінійності знайдено деякі аналітичні оцінки із застосуванням коефіцієнтів Уолша. Зокрема, встановлено, що:

$$NL_f = 2^{n-1} - \frac{1}{2}\max_a\left|W_f(a)\right|,$$

та, відповідно,

$$0 < NL_f \le 2^{n-1} - 2^{n/2-1}$$

Функцію називають *максимально нелінійною*, якщо її нелінійність досягає максимально можливого значення. Виявилось, що клас максимально нелінійних булевих функцій співпадає із класом так званих *бент-функцій*. Бент-функція – це така булева функція, всі коефіцієнти Уолша якої дорівнюють $\pm 2^{n/2}$.

Для багатовимірних булевих функцій висуваються більш суворі вимоги щодо нелінійності. Зокрема, виділяють *сильно нелінійні булеві функції* – це такі $(n,m)-F$ функції, всі координатні функції якої є нелінійними. Виявилось, що всі сильно нелінійні функції мають строгий лавинний ефект нульового порядку; однак чим вища нелінійність функції, тим нижче її кореляційний імунітет. Пряма залежність між нелінійностю та кореляційним імунітетом, яка б дозволяла оцінювати ці параметри в сукупності, поки що не встановлена.

2.4 Лавинні ефекти, коефіцієнти розповсюдження помилок та критерії поширення

Розглянемо булеву функцію $n-f$ та набір векторів e_i, в кожному з яких *i*-тий біт дорівнює 1, а всі інші – 0.

Коефіцієнтом розповсюдження помилки по i-тій змінній булевої функції називають величину:

$$K_i(f) = \sum_x f(x) \oplus f(x \oplus e_i),$$

де сумування ведеться в звичайному арифметичному сенсі.

Коефіцієнти розповсюдження помилки показують, як реагує функція на зміну одного вхідного значення.

Булева функція $n-f$ має *строгий лавинний ефект нульового порядку* (або просто *строгий лавинний ефект*), якщо всі її коефіцієнти розповсюдження помилок по кожній змінній дорівнюють 2^{n-1}.

Булева функція $(n,m)-F$ має *строгий лавинний ефект нульового порядку* (або просто *строгий лавинний ефект*), якщо кожна її координатна функція має строгий лавинний ефект.

Булева функція $(n,m)-F$ має *строгий лавинний ефект k-того порядку*, якщо кожна її $(n-k)$-підфункція, одержана фіксацією довільних k вхідних змінних довільними значеннями, має строгий лавинний ефект.

Власне, можна сказати, що для функції зі строгим лавинним ефектом зміна одного вхідного біту приводить до непередбачуваних змін у значеннях виходу: кожен біт виходу змінюється із ймовірністю 0,5. Тому лавинні ефекти протидіють аналізу функцій за допомогою спостережень за змінами вхідних значень та відповідних їм значень виходу, на яких побудовані диференціальні атаки. Також ці властивості дуже важливі для побудови геш-функцій та генераторів псевдовипадкових чисел.

Умова строгого лавинного ефекту для $(n,m)-F$ булевої функції є доволі суворою. Часто для багатовимірних булевих функцій розглядають більш м'який варіант лавинних ефектів – зміни в середньому. В цьому випадку коефіцієнти розповсюдження помилок визначаються як

$$K_i(F)=\sum_x wt(F(x)\oplus F(x\oplus e_i)),$$

тобто вони відслідковують сумарну кількість змін по всіх координатах.

Булева функція $(n,m)-F$ має *строгий лавинний ефект в середньому*, якщо всі її коефіцієнти розповсюдження помилок дорівнюють половині від максимально можливого значення, тобто $m\cdot 2^{n-1}$. В цьому випадку можна стверджувати, що зміна одиничного біту на вході приводить до зміни в середньому половини бітів виходу, однак імовірність зміни кожного біту виходу не обов'язково дорівнює 0,5.

Строгі лавинні критерії формулюють вимоги до булевих функцій на рівні окремих змінних. Однак на практиці необхідно аналізувати не лише змінні, але й їх можливі комбінації; для цього розглядаються більш загальні критерії поширення.

Похідною за напрямком a одновимірної булевої функції $n-f$ називається булева функція $D_a f(x)=f(x)\oplus f(x\oplus a)$.

Булева функція *f задовольняє критерію поширення за напрямком a*, якщо її похідна за напрямком a є збалансованою: $wt(D_a f)=2^{n-1}$. Множину всіх векторів, для яких функція задовольняє критерію поширення, позначають через PC_f.

Булева функція задовольняє *критерію поширення рівня k*, якщо вона задовольняє критерію поширення за всіма напрямками *a,* для яких $1\le wt(a)\le k$.

Якщо булева функція задовольняє критерію поширення рівня *k*, а вхідні значення розподілені рівноімовірно, то зміна будь-яких *k* вхідних значень призведе до зміни значення виходу функції із імовірністю ½. Ця властивість дуже корисна для побудови надійних поточних шифрів, криптографічних геш-функцій та генераторів псевдовипадкових послідовностей.

Зручним інструментом для аналізу критеріїв поширення виявилась так звана функція автокореляції. *Функція автокореляції* булевої функції $f\in BF_n$ визначається як

$$\Delta_f(u)=\sum_{x\in V_n}(-1)^{f(x)\oplus f(x\oplus u)}=\sum_{x\in V_n}(-1)^{D_u f(x)}.$$

Для функції автокореляції справедливі співвідношення $\Delta_f(0) = 2^n$ та для $u \neq 0$ $\Delta_f(u) = 2^n - 2wt(D_u f)$. Відповідно, f задовольняє критерію поширення за напрямком u тоді та тільки тоді, коли $\Delta_f(u) = 0$; множину PC_f можна визначити як $PC_f = \{u \in V_n : \Delta_f(u) = 0\}$.

Ротхауз довів, що клас функцій, які задовольняють усім можливим критеріям поширення (тобто $PC_f = V_n \setminus \{0\}$), співпадає із класом бент-функцій.

Зручність використання функції автокореляції полягає у тому, що її можна легко обчислити зі спектру Уолша за допомогою перетворення Фур'є, оскільки справедливі співвідношення

$$\Delta_f(v) = \frac{1}{2^n}\sum_{u \in V_n}(-1)^{u \cdot v} W_f^2(u), \qquad W_f^2(u) = \sum_{v \in V_n}(-1)^{u \cdot v}\Delta_f(v).$$

Таким чином, для того, щоб одержали усі значення функції автокореляції, достатньо застосувати швидке перетворення Фур'є до вектору $W = [W_f^2(0),...,W_f^2(2^n - 1)]^T$.

2.6 Диференціальні імовірності

Похідною за напрямком a булевої функції $(n,m) - F$ аналогічно називається булева функція $D_a F(x) = F(x) \oplus F(x \oplus a)$. Для криптоаналізу блокових шифрів важливі випадки, коли похідна багатовимірної булевої функції приймає фіксоване значення із великою імовірністю; це дозволяє провадити декілька потужних криптоаналітичних атак.

Диференціалом булевої функції $(n,m) - F$ називається пара двійкових векторів (a,b), для якої існує таке *x*, що $D_a F(x) = b$. *Імовірністю диференціала* (a,b) називається величина

$$DP^F(a,b) = \Pr_x\{D_a F(x) = b\} = \frac{1}{2^n}\sum_{x \in V_n}\delta(D_a F(x), b),$$

де $\delta(x,y)$ – альтернативний запис дельти Кронекера. Сукупність імовірностей усіх диференціалів називають *диференціальними імовірностями* булевої функції.

В якості основного криптографічного параметру виступає максимальна диференціальна імовірність: $MDP(F) = \max_{a \neq 0, b} DP^F(a,b)$.

3 Виконання операцій у скінченних полях характеристики 2

У завданнях підвищеної складності даної розрахунковій роботі для представлення булевих функцій використовуються обчислення в полях $GF(2^n)$ у поліноміальному базисі. Елементи поля $GF(2^n)$ в цьому випадку являють собою поліноми степеня, що не перевищує $n-1$, над $GF(2)$ або, що те саме, двійкові вектори довжини *n*, які складаються з коефіцієнтів означених поліномів.

Для обчислень у поліноміальному базисі використовується поліном-генератор $p(x)$ – двійковий поліном степеня *n*, що є незвідним над $GF(2)$. Всі операції в полі задаються за модулем $p(x)$.

Додавання у $GF(2^n)$ є звичайним додаванням поліномів над $GF(2)$, що відповідає покомпонентному додаванню за модулем 2 відповідних векторів.

Зведення полінома $q(x)$ за модулем $p(x)$ можна виконати за таким алгоритмом:

1. Поки $\deg(q) \geq n$ виконати:
   1.1. $k = \deg(q)$;
   1.2. $q(x) = q(x) \oplus p(x) \cdot x^{k-n}$.

Зауважимо, що операція $p(x) \cdot x^{k-n}$ фактично є бітовим зсувом двійкового вектора, що відповідає $p(x)$, на $(k-n)$ позицій вліво.

При множенні елементів $GF(2^n)$ відповідні їм поліноми перемножуються, з наступним зведенням результату за модулем незвідного полінома $p(x)$.

Обчислення квадрату в поліноміальному базисі $GF(2^n)$ простіше робити через множення. Для піднесення до степеня (тобто обчислення виразів a^m, де a – елемент поля, а m – натуральне число) можна використати схему Горнера.

Нехай $m = 2^s m_s + 2^{s-1} m_{s-1} + ... + 2m_1 + m_0$. Тоді елемент $b = a^m$ обчислюється за таким алгоритмом:

1. $b := 1$, $c := a$.
2. Для кожного i від 0 до s виконати:
   2.1. Якщо $m_i = 1$, то $b := b \cdot c \bmod p(x)$.
   2.2. $c := c \cdot c \bmod p(x)$.

(Зауважимо, що ми навели лише одну з можливих схем Горнера для обчислення степенів).

При обчисленні степенів за наведеним алгоритмом проміжні результати не будуть перевищувати довжини 2*n* бітів (а при деяких хитрощах при виконанні множення за модулем – то й $(n+1)$-го біта).

Приклад обчислень

Для побудови $GF(2^3)$ використаємо примітивний поліном $f(x) = x^3 + x + 1$ (зауважимо, що ми для зручності позначаємо символом + додавання за модулем два, а не звичайне арифметичне додавання).

Елементами $GF(2^3)$ є поліноми 0, 1, x, $x+1$, x^2, x^2+1, x^2+x, x^2+x+1 або, інакше, відповідні їм вектори (000), (001), (010), (011), (100), (101), (110), (111).

Приклад додавання: $(011) + (101) = (110)$.

Приклад множення: $(011) \cdot (101) = (x+1)(x^2+1) \bmod f(x)$

$$= (x^3 + x^2 + x + 1) \bmod (x^3 + x + 1) = x^2 = (100).$$

4 Швидке перетворення Фур'є

Як можна зауважити, безпосереднє обчислення всього спектру Фур'є булевої функції $n-f$ за визначенням потребує $O(2^n \times 2^n)$ операцій для кожної координатної функції. Така складність виявляється надмірною навіть для функцій невеликого (з точки зору криптографії) розміру. Однак виявляється, що повний спектр Фур'є функції $n-f$ може бути обчислений всього за $O(n2^n)$ операцій за допомогою так званого *швидкого перетворення Фур'є* (ШПФ),

який ми подамо у варіанті, запропонованому Френком Ятсом. Цей метод використовує ітеративну побудову матриць Адамара, які лежать в основі дискретного перетворення Фур'є (та перетворення Уолша-Адамара).

Швидке перетворення Фур'є можна представити у вигляді такого алгоритму.

Ми використовуємо масив $C[u]$, $u \in \{0,1\}^n$, в якому наприкінці роботи будуть зберігатись значення коефіцієнтів Фур'є.

1. (Ініціалізація) Для кожного u встановити $C[u] := f(u)$.
2. Для кожного i від 1 до n виконати:
   a. Розбити всі елементи простору $\{0,1\}^n$ на пари векторів (u_0, u_1), які відрізняються лише значенням i-того біту. Нехай i-тий біт u_0 дорівнює 0, а i-тий біт u_1 дорівнює 1.
   b. Для кожної пари (u_0, u_1) виконати перетворення: в елемент масиву $C[u_0]$ помістити значення $C[u_0] + C[u_1]$, а в елемент масиву $C[u_1]$ – значення $C[u_0] - C[u_1]$.
3. Для кожного u розділити $C[u]$ на нормуючий множник 2^n

Приклад роботи алгоритму наведено у Таблиці 1 (на останньому кроці можна побачити, що знаки доданків утворюють матрицю Адамара розміру 8). Запозичений з Вікіпедії рисунок 2 графічно ілюструє ітеративний принцип роботи алгоритму на кожному кроці.

Швидке перетворення Фур'є легко перероблюється на швидке перетворення Уолша. Дійсно, легко побачити, що перетворення Уолша булевої функції $f(x)$ – це перетворення Фур'є цілочисельної функції $g(x) = 2^n \cdot (-1)^{f(x)}$. Отже, алгоритм ШПФ перетворюється на алгоритм ШПУ після двох модифікацій:

1) Ініціалізація масиву $C[u]$ відбувається таким чином: $C[u] := (-1)^{f(u)}$.
2) Наприкінці алгоритму не потрібно ділити елементи масиву на нормуючий множник 2^n.

Таблиця 1. Приклад роботи алгоритму ШПФ ($n = 3$; символ y_{ijk} дорівнює значенню функції $f(i, j, k)$)

Старт	Перший крок	Другий крок	Третій крок
y_{000}	$y_{000} + y_{100}$	$y_{000} + y_{100} + y_{010} + y_{110}$	$y_{000} + y_{100} + y_{010} + y_{110} + y_{001} + y_{101} + y_{011} + y_{111}$
y_{001}	$y_{001} + y_{101}$	$y_{001} + y_{101} + y_{011} + y_{111}$	$y_{000} + y_{100} + y_{010} + y_{110} - y_{001} - y_{101} - y_{011} - y_{111}$
y_{010}	$y_{010} + y_{110}$	$y_{000} + y_{100} - y_{010} - y_{110}$	$y_{000} + y_{100} - y_{010} - y_{110} + y_{001} + y_{101} - y_{011} - y_{111}$
y_{011}	$y_{011} + y_{111}$	$y_{001} + y_{101} - y_{011} - y_{111}$	$y_{000} + y_{100} - y_{010} - y_{110} - y_{001} - y_{101} + y_{011} + y_{111}$
y_{100}	$y_{000} - y_{100}$	$y_{000} - y_{100} + y_{010} - y_{110}$	$y_{000} - y_{100} + y_{010} - y_{110} + y_{001} - y_{101} + y_{011} - y_{111}$
y_{101}	$y_{001} - y_{101}$	$y_{001} - y_{101} + y_{011} - y_{111}$	$y_{000} - y_{100} + y_{010} - y_{110} - y_{001} + y_{101} - y_{011} + y_{111}$
y_{110}	$y_{010} - y_{110}$	$y_{000} - y_{100} - y_{010} + y_{110}$	$y_{000} - y_{100} - y_{010} + y_{110} + y_{001} - y_{101} - y_{011} + y_{111}$
y_{111}	$y_{011} - y_{111}$	$y_{001} - y_{101} - y_{011} + y_{111}$	$y_{000} - y_{100} - y_{010} + y_{110} - y_{001} + y_{101} + y_{011} - y_{111}$

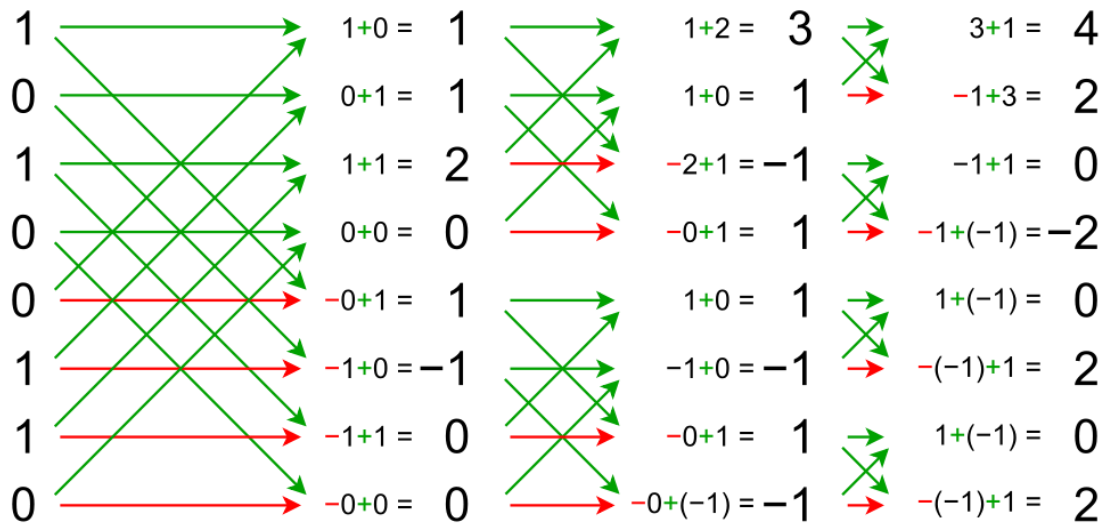

Рисунок 2 – Ілюстрація роботи швидкого перетворення Фур'є на прикладі функції *f*, вектор значень якої дорівнює $(1, 0, 1, 0, 0, 1, 1, 0)$.

5 Знаходження алгебраїчної нормальної форми

Представлення булевої функції у вигляді канонічного поліному (тобто пошук множини коефіцієнтів a_* цього поліному) в загальному випадку виконується за допомогою так званого *перетворення Мебіуса*. Для його опису потрібно ввести деякі додаткові позначення.

Нехай $\bar{u} = (u_1, ..., u_n)$, $\bar{v} = (v_1, ..., v_n)$. Кажуть, що *вектор* $\bar{u}$ *домінує над вектором* $\bar{v}$, якщо виконується умова: $\forall k: u_k \geq v_k$. Для двійкових векторів можна сказати, що вектор $\bar{u}$ домінує над вектором $\bar{v}$, якщо одиниці у векторі $\bar{v}$ стоять лише на тих позиціях, на яких вони стоять у векторі $\bar{u}$ (але не навпаки). Відношення домінування позначається символом $\bar{u} \succ \bar{v}$; легко показати, що домінування є частковим порядком на множині векторів.

Коефіцієнт $a_{i_1,...i_k}$ полінома Жегалкіна булевої функції F будемо позначати як $a_{\bar{u}}$, де вектор $\bar{u}$ має одиниці на місцях $i_1, ..., i_k$ і нулі на всіх інших місцях. Тоді має місце рівність $a_{\bar{u}} = \bigoplus_{\bar{x} \prec \bar{u}} F(\bar{x})$.

Перехід від множини значень функції $\{F(\bar{x})\}$ до множини коефіцієнтів $\{a_{\bar{u}}\}$ описаним вище чином називається *перетворенням Мебіуса*. Перетворення Мебіуса інволютивне: якщо застосувати його до множини $\{a_{\bar{u}}\}$, одержимо множину $\{F(\bar{x})\}$.

Наприклад, нехай $n = 4$; знайдемо коефіцієнт a_{1101} при мономі $x_1x_2x_4$ АНФ булевої функції $f(x_1, x_2, x_3, x_4)$. Вектор 1101 домінує над векторами 0000, 1000, 0100, 0001, 1100, 1001, 0101 та 1101, тому маємо:

$$a_{1101} = f(0000) \oplus f(1000) \oplus f(0100) \oplus f(0001) \oplus f(1100) \oplus f(1001) \oplus f(0101) \oplus f(1101).$$

Складність виконання перетворення Мебіуса для одержання всієї множини $\{a_{\bar{u}}\}$ дорівнює $O(3^n)$ додавань (покажіть це!); однак при безпосередній реалізації така складність вимагатиме досить важких організаційних витрат (побудова відношення домінування на векторах тощо). Перевагою цього методу є можливість обчислення кожного коефіцієнта окремо.

Існує інший метод обчислення коефіцієнтів полінома Жегалкіна в сукупності, який багато в чому подібний до швидкого перетворення Фур'є. За аналогією ми будемо називати його *швидким перетворенням Мебіуса* (ШПМ).

Ми використовуємо масив $A[u]$, $u \in \{0,1\}^n$, в якому наприкінці роботи будуть зберігатись значення коефіцієнтів поліному Жегалкіна.

1. (Ініціалізація) Для кожного u встановити $A[u] := f(u)$.
2. Для кожного i від 1 до n виконати:
   a. Розбити всі елементи простору $\{0,1\}^n$ на пари векторів (u_0, u_1), які відрізняються лише значенням i-того біту. Нехай i-тий біт u_0 дорівнює 0, а i-тий біт u_1 дорівнює 1.
   b. Для кожної пари (u_0, u_1) виконати перетворення: в елемент масиву $A[u_1]$ помістити значення $A[u_0] \oplus A[u_1]$.

Приклад роботи алгоритму наведено у Таблиці 2.

Таблиця 2 – Приклад роботи алгоритму ШПМ ($n = 3$; символ y_{ijk} дорівнює значенню функції $f(i, j, k)$)

Старт	Перший крок	Другий крок	Третій крок
y_{000}	y_{000}	y_{000}	y_{000}
y_{001}	y_{001}	y_{001}	$y_{000} \oplus y_{001}$
y_{010}	y_{010}	$y_{000} \oplus y_{010}$	$y_{000} \oplus y_{010}$
y_{011}	y_{011}	$y_{001} \oplus y_{011}$	$y_{000} \oplus y_{010} \oplus y_{001} \oplus y_{011}$
y_{100}	$y_{000} \oplus y_{100}$	$y_{000} \oplus y_{100}$	$y_{000} \oplus y_{100}$
y_{101}	$y_{001} \oplus y_{101}$	$y_{001} \oplus y_{101}$	$y_{000} \oplus y_{100} \oplus y_{001} \oplus y_{101}$
y_{110}	$y_{010} \oplus y_{110}$	$y_{000} \oplus y_{100} \oplus y_{010} \oplus y_{110}$	$y_{000} \oplus y_{100} \oplus y_{010} \oplus y_{110}$
y_{111}	$y_{011} \oplus y_{111}$	$y_{001} \oplus y_{101} \oplus y_{011} \oplus y_{111}$	$y_{000} \oplus y_{100} \oplus y_{010} \oplus y_{110} \oplus y_{001} \oplus y_{101} \oplus y_{011} \oplus y_{111}$

Порядок виконання розрахункової роботи

Під час роботи необхідно дослідити криптографічні властивості наданих булевих функцій відповідно до варіанту завдання. Студент **обов'язково виконує основний варіант** та за бажанням – варіант підвищеної складності.

У основному варіанті для дослідження пропонується одномірна булева функція від п'яти змінних, для якої необхідно:

а) побудувати таблицю істинності;

б) знайти спектр Уолша, використовуючи швидке перетворення Уолша;

в) знайти алгебраїчну нормальну форму, використовуючи швидке перетворення Мебіуса, та значення алгебраїчного степеня;

г) обчислити значення функції автокореляції;

ґ) вказати, чи є серед вхідних змінних несуттєві;

д) знайти значення дисбалансу;

е) знайти значення нелінійності та знайти всі найкращі лінійні статистичні аналоги даної функції;

є) знайти рівень кореляційного імунітету;

ж) знайти усі вектори, за якими функція задовольняє критеріям поширення, та знайти рівень критеріїв поширення;

з) обчислити коефіцієнт розповсюдження помилки за змінною, номер якої дорівнює $(V \bmod 5)+1$, де V – номер варіанту, та зробити висновок про наявність строгого лавинного ефекту;

и) зробити загальний висновок про можливість використання запропонованої функції у криптографічних цілях.

Виконання даних завдань не передбачає використання комп'ютерної техніки та навичок програмування. Робота повинна містити всі проміжні розрахунки, зокрема, покрокові обчислення ШПУ та ШПМ.

Варіанти звичайних завдань наведено у Таблиці 3.

У варіантах завдань підвищеної складності для дослідження пропонуються дві багатомірні булеві функції: $f(x) = x^N$ та $g(x) = x^M$, де x – елемент скінченного поля $GF(2^n)$, N та M – задані натуральні числа і всі обчислення виконуються у полі $GF(2^n)$, побудованому за допомогою полінома-генератора $p(x)$. Відповідно, обидва досліджувані перетворення є (n,n)-булевими функціями.

Для досліджень пропонуються перетворення, визначені над полями $GF(2^{15})$, $GF(2^{16})$ та $GF(2^{17})$ у поліноміальному базисі. В якості поліномів-генераторів використовуються такі поліноми:

$GF(2^{15})$: $p(x) = x^{15} + x + 1$,

$GF(2^{16})$: $p(x) = x^{16} + x^{12} + x^3 + x + 1$

$GF(2^{17})$: $p(x) = x^{17} + x^3 + 1$.

Для функцій $f(x)$ та $g(x)$ необхідно визначити такі криптографічні властивості:

а) для кожної координатної функції:

– дисбаланс,

– алгебраїчний степінь,

– нелінійність,

– рівень кореляційного імунітету,

– коефіцієнти розповсюдження помилок (по кожній змінній),

– відносне відхилення у процентах коефіцієнтів розповсюдження помилок від середнього значення (тобто величину $\varepsilon_i = \left|K_i(f) - 2^{n-1}\right| / 2^{n-1}$, точність до другого знаку).

б) для булевої функції в цілому:

– алгебраїчний степінь,

– коефіцієнти розповсюдження помилок в середньому (по кожній змінній),

– відносне відхилення у процентах коефіцієнтів розповсюдження помилок в середньому (тобто величину $\varepsilon_i = \left|K_i(f) - n \cdot 2^{n-1}\right| / (n \cdot 2^{n-1})$, точність до другого знаку),

– наявність лавинного ефекту нульового рівня,

– наявність строгого лавинного ефекту в середньому,

– максимум диференціальної імовірності.

Виконання завдань підвищеної складності передбачає використання комп'ютерних обчислень. Всі програми, створені для розрахунків, надаються викладачеві окремо в електронному вигляді. Варіанти завдань підвищеної складності наведено у таблиці 4.

Методичні вказівки до виконання завдань підвищеної складності

При обчисленні криптографічних характеристик «в лоб» ви будете змушені виконати великий об'єм обчислень. Намагайтесь оптимізувати обчислення, аби не витрачати час. Оскільки розмір входу функцій є відносно невеликим, то роботу може сильно пришвидшити попереднє обчислення таблиць істинності досліджуваних функцій та спектру коефіцієнтів Уолша; для обчислення останніх може виявитись корисним попереднє обчислення скалярних добутків векторів потрібної довжини та/або реалізація швидкого перетворення Уолша. Лавинні ефекти (за деякої «акробатичності» в програмуванні) можна оцінювати для всіх координатних функцій водночас. Так само диференціальні імовірності при фіксованому значенні a можуть бути обчислені для всіх значень b водночас, оскільки фактично нас цікавить розподіл вихідних значень похідної $D_a F(x)$. Існують й інші можливості для пришвидшення обчислень.

Однак пам'ятайте, що на першому місці повинна стояти коректність обчислень, а вже після неї стоїть швидкість.

Для виконання роботи ви можете використовувати будь-яку мову програмування, однак пам'ятайте, що інтерпретаторні мови п'ятого покоління (такі як Perl та Python) на порядок повільніші за мови нативні (C/C++ та Pascal/Delphi). Платформонезалежні мови четвертого покоління (Java, C#, Scala, Go тощо) швидші за інтерпретаторні, але також поступаються нативним.

Таблиця 3 – Варіанти звичайних завдань (№ – номер варіанту)

№	$f(x_1, x_2, x_3, x_4, x_5)$	№	$f(x_1, x_2, x_3, x_4, x_5)$
1	$x_1 \oplus (x_2 x_3 \vee (x_4 \to x_5))$	31	$x_1 \bar{x}_2 x_3 \vee x_4 x_5$
2	$x_1 \vee x_2 x_3 \vee (x_4 \sim x_5)$	32	$x_1 \vee (x_2 \to \bar{x}_3 x_4) x_5$
3	$(x_1 \sim x_2) \oplus (x_3 \sim x_4) \oplus (x_4 \sim x_5)$	33	$(x_1 \oplus \bar{x}_2) \vee (x_3 x_4 \oplus \bar{x}_5)$
4	$(x_1 \vee x_2) \sim (x_3 \vee x_4 \vee x_5)$	34	$(\bar{x}_1 x_2 x_3 \to x_4) \to \bar{x}_5$
5	$(x_1 x_2 \to x_3 x_4) \to x_5$	35	$\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \oplus x_5$
6	$(x_1 \sim x_2) x_3 \oplus (x_4 \sim x_5) x_1$	36	$x_1 \sim (x_2 \sim (\bar{x}_3 \sim (x_4 \sim x_5)))$
7	$(x_1 \sim x_2) x_3 \oplus (x_1 \sim x_3) x_2 \oplus x_4 x_5$	37	$((x_1 \to \bar{x}_2) \vee x_1 x_3) \oplus (x_2 \bar{x}_4 \vee x_5)$
8	$x_1 x_2 \oplus x_3 \oplus (x_4 \sim x_5)$	38	$x_1 x_2 \vee \bar{x}_3 x_4 \vee x_5 x_1$
9	$(x_1 \to \bar{x}_2) \bar{x}_3 \vee (x_4 \to \bar{x}_5)$	39	$\bar{x}_1 x_3 \oplus \bar{x}_4 \oplus x_5 x_1$
10	$(x_1 \sim x_2) \to (x_4 \sim x_5)$	40	$(x_1 \bar{x}_2 \to x_3 x_4) \sim \bar{x}_5 x_3$
11	$x_1 x_2 x_3 \vee x_4 x_5$	41	$x_1 \oplus (x_2 \vee x_3 \vee (x_4 \to x_5))$
12	$x_1 \vee (x_2 \to x_3 x_4) x_5$	42	$x_1 \vee (x_2 \sim x_3) \vee (x_4 \sim x_5)$
13	$(x_1 \oplus x_2) \vee (x_3 x_4 \oplus x_5)$	43	$(\bar{x}_1 \sim x_2) \oplus x_3 x_4 \oplus (x_4 \sim x_5)$
14	$(x_1 x_2 x_3 \to x_4) \to x_5$	44	$(x_1 \vee \bar{x}_2) \oplus (x_3 \vee \bar{x}_4 \vee x_5)$
15	$x_1 x_2 x_3 x_4 \oplus x_5$	45	$(x_1 \bar{x}_2 \to x_3 x_4) \sim x_5$
16	$x_1 \sim (x_2 \sim (x_3 \sim (x_4 \sim x_5)))$	46	$(x_1 \sim x_3) x_2 \oplus (x_4 \sim x_5) x_3$
17	$((x_1 \to x_2) \vee x_1 x_3) \oplus (x_2 x_4 \vee x_5)$	47	$(x_1 \sim x_3) x_2 \oplus (\bar{x}_1 \sim x_3) x_4 \oplus x_4 x_5$
18	$x_1 x_2 \vee x_3 x_4 \vee x_5 x_1$	48	$(\bar{x}_1 x_2 \sim x_3) \oplus (x_4 \sim x_5)$
19	$x_1 x_3 \oplus x_4 \oplus x_5 x_1$	49	$(x_1 \to \bar{x}_3) x_2 \vee (\bar{x}_4 \to \bar{x}_5)$
20	$(x_1 x_2 \to x_3 x_4) \sim x_5 x_3$	50	$(x_1 \oplus x_2) \to (x_4 \oplus x_5)$
21	$x_1 \oplus (x_2 \bar{x}_3 \vee (\bar{x}_4 \to x_5))$	51	$x_1 x_2 \bar{x}_3 \oplus x_4 \bar{x}_5$
22	$(x_1 \vee \bar{x}_2) \sim (\bar{x}_3 \vee x_4 \vee x_5)$	52	$\bar{x}_1 \vee (\bar{x}_2 \to x_3 x_4) x_5$
23	$\bar{x}_1 \vee x_2 x_3 \vee (x_4 \sim \bar{x}_5)$	53	$(x_1 \sim x_2) \vee (x_3 x_4 \sim x_5)$
24	$(x_1 \sim \bar{x}_2) \oplus (x_3 \sim \bar{x}_4) \oplus (x_4 \sim \bar{x}_5)$	54	$(x_1 x_2 x_3 \oplus x_4) \to x_1 x_5$
25	$(x_1 x_2 \to \bar{x}_3 x_4) \to x_5$	55	$x_1 x_2 x_3 \oplus x_4 \oplus x_5$
26	$(x_1 \sim x_2) \bar{x}_3 \oplus (x_4 \sim x_5) \bar{x}_1$	56	$\bar{x}_1 \sim (x_2 \sim (x_3 \sim (x_4 \oplus x_5)))$
27	$(x_1 \sim \bar{x}_2) x_3 \oplus (x_1 \sim \bar{x}_3) x_2 \oplus x_4 x_5$	57	$((\bar{x}_1 \to x_2) \vee x_1 x_3) \sim (x_2 x_4 \vee \bar{x}_5)$
28	$x_1 x_2 \oplus \bar{x}_3 \oplus (\bar{x}_4 \sim x_5)$	58	$\bar{x}_1 x_2 \vee \bar{x}_3 x_4 \vee \bar{x}_5 x_1$
29	$(\bar{x}_1 \to \bar{x}_2) x_3 \vee (x_4 \to \bar{x}_5)$	59	$\bar{x}_1 x_3 \oplus (x_4 \sim x_5 x_1)$
30	$(\bar{x}_1 \sim x_2) \to (\bar{x}_4 \sim x_5)$	60	$(\bar{x}_1 \bar{x}_2 \to x_3 x_4) \oplus x_5 x_2$

Таблиця 4 – Варіанти завдань підвищеної складності (№ – номер варіанту, n – степінь розширення поля, N та M – експоненти для дослідження)

№	*n*	*N*	*M*	№	*n*	*N*	*M*
1	15	32766	32765	21	15	5	4
2	16	65534	65533	22	17	5	6
3	17	131070	131069	23	15	17	18
4	15	131	132	24	17	17	16
5	17	259	260	25	15	129	128
6	15	2175	2176	26	17	9	10
7	17	271	270	27	15	257	256
8	15	4679	4680	28	17	33	32
9	15	3	2	29	15	2049	2048
10	16	3	2	30	17	65	64
11	17	3	2	31	15	8193	8194
12	15	13	12	32	17	129	130
13	16	57	56	33	15	16385	16386
14	17	13	14	34	17	257	258
15	15	241	240	35	17	513	512
16	16	993	990	36	17	1025	1026
17	17	57	60	37	17	2049	2050
18	15	16257	16256	38	17	4033	4032
19	16	16257	16258	39	17	16257	16260
20	17	993	994	40	17	65281	65280

Вимоги до оформлення розрахункової роботи

Розрахункова робота виконується на листах А4 або у стандартних зошитах в клітинку (для основних варіантів завдань); титульний лист оформлюється за зразком, наведеним у Додатку А.

Розрахункова робота, виконана за основним завданням, повинна містити:
- завдання розрахункової роботи;
- таблицю істинності досліджуваної функції (включно із проміжними розрахунками);
- спектр Уолша досліджуваної функції (включно із розрахунками швидкого перетворення Уолша);
- алгебраїчну нормальну форму досліджуваної функції, представлену у вигляді канонічного поліному (включно із розрахунками швидкого перетворення Мебіуса);
- чисельні значення криптографічних параметрів досліджуваної функції: дисбаланс, нелінійність, рівень кореляційного імунітету, коефіцієнт розповсюдження помилки, разом із шляхами їх одержання;
- перелік найкращих лінійних статистичних аналогів (із обґрунтуванням, чому вони найкращі);
- аналіз та інтерпретацію одержаних чисельних значень з точки зору криптографії;
- загальні висновки.

Розрахункова робота, виконана за завданням підвищеної складності, повинна містити:
- завдання розрахункової роботи;

– чисельні значення криптографічних параметрів булевих функцій та їх координатних функцій, подані у вигляді таблиць;
– аналіз та інтерпретацію одержаних чисельних значень з точки зору криптографії (включаючи час роботи, який витратив ваш програмний код на обчислення тих чи інших параметрів);
– загальні висновки.

Графік виконання розрахункової роботи

Виконання розрахункової роботи включає в себе три етапи:
1) виконання розрахункової роботи;
2) написання програмного коду та демонстрація його викладачеві (для завдань підвищеної складності);
3) оформлення розрахункової роботи та здача його у встановлений термін;
4) теоретичний захист розрахункової роботи.

Терміни виконання РР встановлюються викладачами після видачі завдання.

Оцінювання розрахункової роботи

За виконання розрахункової роботи студент може одержати до 10 рейтингових балів, що включають в себе бали за самостійну роботу, теоретичний захист та додаткові організаційні бали. **Виконання основного завдання є обов'язковим.** При виконанні окрім основного завдання також завдання підвищеної складності студент може одержати додаткові бали за написання відповідних програм (до 5 балів), які враховуються як бонусні бали у семестровому рейтингу.

Зауважимо, що виконання даної розрахункової роботи є необхідною умовою допуску до заліку.

Критерії оцінювання розрахункової роботи:
– всі чисельні значення одержано правильно – 4 бали;
– робота містить незначну кількість арифметичних помилок, які викликані неуважністю та не впливають на висновки – 2-3 бали;
– робота містить грубі помилки, але вірні ідеї обчислень – 1 бал;
– робота не виконана – 0 балів, недопуск до заліку

Критерії оцінювання теоретичного захисту:
– повне розуміння теоретичного матеріалу, чіткі відповіді на всі питання – 5 балів;
– окремі недоліки при відповідях на питання – 3-4 бали;
– неповне розуміння теоретичного матеріалу, грубі помилки – 1-2 бали;
– нерозуміння теоретичного матеріалу, відсутність відповідей на питання – 0 балів.

Додаткові організаційні бали включають в себе:
– виконання розрахункової роботи у встановлений термін – 1 бал;
– невиконання розрахункової роботи у встановлені терміни: (–1) бал за кожну добу запізнення.

Критерії оцінювання програмного коду (для завдань підвищеної складності):
– код написано повністю правильно, розрахункові результати вірні – 5 балів;
– код працює із помилками, не всі розрахунки вірні – 1-4 бали;
– код працює некоректно, всі розрахунки невірні – 0 балів.

Несвоєчасна здача розрахункової роботи штрафується на (–1) бал за кожну добу після встановленого терміну здачі робіт.

Програмний код, створений для виконання додаткового завдання підвищеної складності, перевіряється на наявність неправомірних запозичень (плагіату) за допомогою сервісу *Stanford MOSS Antiplagiarism*. У разі виявлення в програмному коді неправомірних запозичень студент замість додаткових балів одержує штраф (–10) балів.

Список рекомендованої літератури

1. Математичні методи захисту інформації. Курс лекцій. Ч І. / Укладачі Завадська Л.О., Савчук М.М. – К.: НТУУ «КПІ», 2008. – 128 с.
2. Харин Ю.С. Математические и компьютерные основы криптологии / Ю.С. Харин, В.И. Берник, Г.В. Матвеев, С.В. Агиевич – Минск: Новое знание, 2003. – 382с.
3. Логачев О.А. Булевы функции в теории кодирования и криптологии / О.А. Логачев, А.А. Сальников, В.В. Ященко – М.: МЦНМО, 2004. – 470с.
4. Фомичев В.М. Дискретная математика и криптология. / В.М. Фомичев – М.: ДИАЛОГ-МИФИ, 2003 – 400с.

ДОДАТОК А
До методичних вказівок

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського”
Фізико-технічний інститут
Кафедра математичних методів захисту інформації

РОЗРАХУНКОВА РОБОТА

з дисципліни
СИМЕТРИЧНА КРИПТОГРАФІЯ

Варіант №_____

Виконав:

студент _____ курсу групи_____

_____
(ПІБ студента)

Прийняв:

Викладач_____
_____
(ПІБ викладача)

Кількість балів:

_____

2019