МЕТОДИ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕХАНІЗМІВ
ЛАБОРАТОРНА РОБОТА №2
“Реалізація алгоритмів генерації ключів гібридних систем”.

Недождій Максим, Буржимський Ростислав

ФІ-42мн

1 Мета роботи

Дослідження алгоритмів генерації псевдовипадкових послідовностей, тестування простоти чисел та генерації простих чисел з точки зору їх ефективності за часом та можливості використання для гененерації ключів асиметричних криптосистем.

2 Постановка задачі

Для першого типу лабораторних дослідити різні методи генерації випадкових послідовностей для засобів обчислювальної техніки. Дослідити ефективність за часом алгоритми тестування на простоту різних груп – імовірнісних, гипотетичних та детермінованих.

Хід роботи:

Робота полягала у написанні генераторів, написанні тестів, перевірки коректності роботи генераторів, перевірки коректності роботи тестів, виправленні помилок. Основною проблемою була відсутність влаштованих бібліотек для роботи за BM та BBS, виникла необхідність завантажувати зовнішні бібліотеки для роботи з великими числами. Необхідність уніфікації виводу усіх генераторів для подальшого тестування також ускладнила завдання.

Результати дослідження:

Довжина послідовності, яка була згенерована для перевірок статистики: 2097152 бітів

Результати дослідження при рівні значущості $\alpha = 0.01$

a1 = 0.01	Рівноімовірність			Незалежність			Однорідність		
	теоретична	практична	пройшов	теоретична	практична	пройшов	теоретична	практична	пройшов
Вбудований генератор	310.457	245.186		65866.9	64705.6		131496	130076	
LehmerLow		0			infinity			0.124512	
LehmerHigh		22.0173			59539.9			127263	
L20		206.852			56812.9			129960	
L89		241.199			64862.1			130258	
Geffe		276.236			79667.4			112916	
<<Бібліотекар>>		2.07*10^7			infinity			infinity	
Вольфрам		565.277			122137			131239	
BM		226.703			65085.8			130262	
BM_bytes		3.28*10^9			infinity			129226	
BBS		269.115			65000.5			130303	
BBS_bytes		272.701			64873.5			130337	

Результати дослідження при рівні значущості $\alpha = 0.05$

a2 = 0.05	Рівноімовірність			Незалежність			Однорідність		
	теоретична	практична	пройшов	теоретична	практична	пройшов	теоретична	практична	пройшов
Вбудований генератор	293.248	293.756		65619.3	64713.3		131146	130420	
LehmerLow		0			infinity			0.124512	
LehmerHigh		20.238			62121.8			128230	
L20		177.877			57124.5			130287	
L89		218.531			65306.1			130086	
Geffe		253.467			80050.2			127269	
<<Бібліотекар>>		2.07*10^7			infinity			infinity	
Вольфрам		564.602			144228			130072	
BM		227.217			65057.9			130092	
BM_bytes		3.28*10^9			infinity			130012	
BBS		223.475			65449.2			130392	
BBS_bytes		287.33			64701.2			130625	

Результати дослідження при рівні значущості $\alpha = 0.1$

a3 = 0.1	Рівноімовірність			Незалежність			Однорідність		
	теоретична	практична	пройшов	теоретична	практична	пройшов	теоретична	практична	пройшов
Вбудований генератор	284.336	277.664		65487.6	65051.5		130960	130857	
LehmerLow		0			infinity			0.124512	
LehmerHigh		21.4504			59807.2			127857	
L20		202.283			57310.6			130368	
L89		269.963			65571.6			130952	
Geffe		255.879			80355.7			127644	
<<Бібліотекар>>		2.07*10^7			infinity			infinity	
Вольфрам		482.521			158138			131782	
BM		264.822			65468.1			130135	
BM_bytes		3.28*10^9			infinity			129568	
BBS		249.666			65199			130720	
BBS_bytes		256.487			65552			131418	

Порівняння генераторів:

Згідно з результатами тестів, нам гарантовано не підходять:

- «Бібліотекар», бо послідовність не випадкова
- Вольфрам, бо результат дуже неякісний
- BM, бо працює довго, не підходить для практичного застосування
- LehmerLow та Geffe не проходять тести на незалежність
- L20 має маленький період, тобто для великих послідовностей будуть з'являтись повтори гарантовано

Усі генератори, окрім BM проходять тести без значних затримок, але, як відомо з курсу симкрипти, Geffe не можна застосувати у криптографічних цілях. LehmerHigh пройшов усі перевірки, але має недостатньо великий період. Вбудований генератор, BBS і L89 пройшли усі перевірки (з точністю до похибки), а отже, їх можна застосовувати для генерування псевдовипадкових послідовностей.

Висновок:

Під час виконання не було повністю зрозуміло, які генератори дійсно не задовольняють властивостей, які вимагають від них тести, а де виникають проблеми з написаним кодом. Однак, очевидно "неробочі" генератори одразу попадались на дуже великих значеннях, і коли було уточнено, що відвертих помилок нема, то проаналізувати вдалось успішно.