УДК 681.324.067

Н.В. ШАПОЧКА

Харківський національний університет радіоелектроніки, Україна

АНАЛІЗ АТАК НА ГЕНЕРАТОРИ ВИПАДКОВИХ БІТІВ

Проводиться аналіз генераторів випадкових бітів (ГВБ). На математичному рівні наведено опис ГВБ. Визначено, що важливою характеристикою ГВБ є їх стійкість до криптоаналітичних атак при оцінці криптографічного алгоритму. Розглянуті можливі атаки на ГВБ, показано, як вони працюють, і який вплив мають на ГВБ. Наведено класифікацію атак на ГВБ. Зроблено аналіз деяких ГВБ на вразливість до описаних атак.

Ключові слова: *генератор випадкових бітів, випадковість, ентропія, статистичні властивості, криптоаналітичні атаки.*

Введення

Поняття випадковості, непередбачуваності, складності і ентропії є концептуальною основою криптографії.

Можна сказати, що створення будь-якої криптосистеми (наприклад, схеми шифрування або обміну ключів) зводиться до розробки такого перетворення, яке було б в необхідному ступені непередбачуваним для зовнішнього спостерігача [1].

Якість генераторів випадкових бітів (ГВБ) в значній мірі впливає на роботу програм, які використовують випадкові біти. Тому до генераторів випадкових бітів висуваються багато складних вимог. Також важливою характеристикою ГВБ є їх стійкість до криптоаналітичних атак при оцінці криптографічного алгоритму. На сьогоднішній день існує багато аналітичних атак, які представляють велику загрозу ГВБ.

Метою даної статті є розгляд можливих атак на ГВБ, а також аналіз деяких ГВБ на вразливість до описаних атак.

Математичне визначення генераторів псевдовипадкових бітів

Вимогою мінімального захисту для генераторів псевдовипадкових бітів (ГПВБ) є вимога достатньо великої довжини k випадкового початкового числа, такої, щоб пошук на 2^k елементах був нездійсненним для зловмисника.

Двома загальними вимогами є:

–вимога невідмінності вихідних послідовностей ГПВБ від істинно випадкових послідовностей;

–вимога непередбаченості вихідних бітів для зловмисника з обмеженими обчислювальними ресурсами [3].

Визначення 1. Вважається, що ГПВБ проходить всі статистичні тестування поліноміального часу, якщо жоден алгоритм поліноміального часу не може правильно відрізнити вихідну послідовність генератора від істинної випадкової послідовності тієї ж довжини з ймовірністю, значно більшою, ніж $1/2$ [3].

Визначення 2. Вважається, що ГПВБ проходить тестування наступного біту, якщо немає жодного алгоритму поліноміального часу, який при введені перших l бітів вихідної послідовності s, може передбачити $(l+1)$–й біт s з ймовірністю, значно більшою, ніж $1/2$ [3].

Теорема 1. ГПВБ проходить тестування наступного біту, якщо він проходить всі статистичні тестування поліноміального часу [3].

Блюм та Мікалі ввели поняття криптографічно стійкого ГПВБ, еквівалентне наступному визначенню [3].

Визначення 3. ГПВБ, який проходить тестування наступного біту, називається криптографічно стійким ГПВБ [3].

Псевдовипадкова послідовність повинна мати деякі статистичні властивості, присутні в істинно випадкових послідовностях. Багато статистичних властивостей лінійної конгруентної псевдовипадкової числової послідовності.

$$x_{i+1} = a \cdot x_i + b \pmod n$$

досліджені Кнутом. Проте, на відміну від істинних випадкових послідовностей, наступний біт в лінійній конгруентній послідовності може бути легко обчислений з попередніх бітів, навіть коли x_0, a, b або n не задані. Хоча лінійні конгруентні генера-

© Н.В. Шапочка

тори проходять деякі статистичні тестування, вони є передбаченими, і тому цілком незахищеними з точки зору криптографії [3].

ГПВБ, що засновані на детермінованих алгоритмах, відповідають наступній структурі, узятій з Ля Екюера (1994): ГПВБ є структурою (S,μ,f,U,g), де S – кінцева множина станів (простір станів), μ – розподіл ймовірності на S станів, що використовується для вибору початкового стану (або початкового числа) s_0, $f: S \to S$ є перехідна функція, U – є вихідний простір, і $g: S \to U$ є вихідна функція. Зазвичай, $U = (0,1)$. Стан ГПВБ формується згідно рекурентності $s_i = f(s_{i-1})$ для $i \geq 1$, а вихідними даними на кроці i є $u_i = g(s_i) \in U$. Вихідні значення $u_0, u_1, u_2 \ldots$ є так званими випадковими бітами, виробленими ГПВБ [4].

Оскільки S є кінцевим, то повинні бути деякі кінцеві $l \geq 0$ і $j > 0$ такі, що $s_{l+j} = s_l$. Тоді для всіх $i \geq l$ мається $s_{i+j} = s_i$ і $u_{i+j} = u_i$. Це відбувається тому, що f і g є детермінованими. Тобто послідовності станів і вихідних даних є періодичними. Найменше позитивне j, для якого цей випадок має місце, називається довжиною періоду ГПВБ і позначається як ρ. Коли $l = 0$, то послідовність вважається чисто періодичною.

Очевидно, $\rho \leq |S|$, потужність S. Якщо стан має k-бітове представлення на комп'ютері, то $\rho \geq 2^k$. Хороші ГПВБ проектуються таким чином, щоб їх довжина періоду не дуже відрізнялася від верхньої межі. Взагалі, значення ρ може залежати від початкового числа s_0, але хороші ГПВБ зазвичай проектуються таким чином, щоб довжина періоду була однаковою для всіх допустимих початкових чисел [4].

Атаки на генератори випадкових бітів

Головною відмінністю між генераторами випадкових бітів для стохастичних моделювань і криптографічними застосуваннями є те, що в криптографічних системах ГВБ додатково повинні забезпечувати стійкість проти атак.

Загалом атаки означають спробу:

– знаходження невідомих вихідних даних;

– знаходження інформації про внутрішній стан (і, значить, про майбутні вихідні дані); або

– маніпулювання вихідними даними генератора.

Можливі атаки можна розділити на три різних класи: криптоаналітичні атаки, атаки, засновані на вхідних даних, і атаки розповсюдження компрометації стану. Перший тип атак робить спробу знайти інформацію про внутрішній стан або майбутні вихідні дані генератора через спостереження частини поточних вихідних даних. Атаки, засновані за вхідних даних, досягають своєї мети через спостереження або маніпулювання вхідними даними ГВБ. Метою останнього типу атаки є розширення знання поточного стану генератора до майбутнього або минулого стану [2].

Розглянемо декілька атак на генератори випадкових бітів, покажемо, як вони працюють, і який вплив мають на ГВБ.

Прямі криптоаналітичні атаки: протягом цієї атаки зловмисник спостерігає вихідні дані і намагається знайти будь-яку інформацію про внутрішній стан або майбутні вихідні дані генератора. Багато ГВБ використовують криптографічні примітиви, подібні геш-функціям (наприклад, SHA-1 або MD5), або блокові шифри (DES, потрійний DES, AES тощо) для запобігання цьому виду атак. Базовим припущенням є перенесення криптографічного захисту примітивів на генератори, які використовують їх.

Проте, як показав аналіз, не доцільно всліпу довіряти генераторам, які будуються на криптографічних примітивах.

Гарним прикладом цього є генератор сеансових ключів Kerberos 4. Спеціальний метод використання примітивів має головний вплив також і на захист генератора. Генератор Kerberos 4 виробляє 56-бітовий ключ для блокового шифру DES за допомогою двох послідовних викликів UNIX випадкової функції, яка використовує тільки 32-бітовий ключ. Випадкова функція переініціалізовується з кожним запитом ключа. Згодом, стійкість шифрування і, значить, стійкість проти криптоаналітичних атак зменшується з 56 до 32 бітів. Для знаходження правильного ключа відкритого тексту-шифротексту через атаку грубої сили витрачається близько 6 годин на DEC Alpha, але, як бачимо, 56 бітова стійкість шифрування є тільки ілюзією. Це є найслабкішою ланкою в ланцюзі, що розраховується [2].

Атака з частковим попереднім обчисленням може застосовуватися до будь-якого генератора, який використовує лічильник. Припустимо, що ніякі вхідні дані не оброблені, і зловмисник здатний спостерігати послідовні вихідні дані. У наступному кроці він повинен обчислити вихідні дані (у нашому випадку, геш-функція) кожного t-го значення лічильника і запам'ятати їх в списку. Одне з t спостережених значень повинно увійти до списку. Після запису вхідних даних у списку, можна знайти внутрішній стан генератора. Всі подальші вихідні дані

залишаються відомими до тих пір, поки не будуть оброблені ніякі нові вхідні дані [2].

Хронометражна атака використовує той факт, що інкрементація лічильника вимагає різної кількості часу, залежно від передбаченої кількості байтових додавань. Якщо зловмисник може виміряти час, необхідний для інкрементації лічильника, він може зробити висновки по числу нулів в поточному стані лічильника. Можливий також варіант вгадування моменту часу, коли всі байти нижчого порядку є нульові, оскільки тоді для попереднього інкременту необхідно особливо багато байтових додавань. Така ситуація позначається як «слабкий стан» в хронометражній атаці. Інформація, що отримується завдяки цій атаці, може бути скомбінована з атакою попередніх обчислень. Це означає, що зловмиснику відоме, коли вигідніше порівнювати вихідні дані зі списком попередніх обчислень [2].

Атаки, засновані на вхідних даних: в цій атаці зловмисник здатний спостерігати або маніпулювати вхідними даними генератора. Метою цієї атаки є зменшення числа можливих вихідних даних для полегшення таким чином вгадування, або навіть змушення генератора виробляти необхідні вихідні дані. Існує три різних види атак, заснованих на вхідних даних – атаки з вибраними вхідними даними, атаки з підробленими вхідними даними і атаки з відомими вхідними даними. Розглянемо різні атаки на декількох прикладах [2].

Атаки з вибраними вхідними даними: зловмисник може безпосередньо маніпулювати вхідними даними генератора. Така ситуація з'являється відносно рідко, але дозволяє реалізувати дуже ефективні атаки [2].

Дослідимо цю атаку на прикладі генератора DSA. Генератор DSA заснований на геш-функції SHA і призначений для вироблення DES ключів. Всі функції додавання відбуваються модулем 2^N, де $160 \le N \le 512$. Нехай $N = 160$, оскільки це значення представляє найслабкішу версію генератора. Генератор містить внутрішній стан X_i, $i \ge 1$. Нове вхідне W_i обробляється кожного разу, коли генеруються вихідні дані. Якщо ніяких вхідних даних немає, то W_i встановлюється в нуль. Вихідні дані генеруються як

$$\text{output}[i] \equiv \text{SHA}\left(W_i + X_i \left(\bmod 2^{160}\right)\right) \text{ i}$$

$$X_{i+1} \equiv X_i + \text{output}[i] + 1 \left(\bmod 2^{160}\right).$$

Ефективна атака встановлюватиме вхідні дані генератора в

$$W_i \equiv W_{i-1} - \text{output}[i-1] - 1 \left(\bmod 2^{160}\right).$$

Ці нові вхідні дані змушують генератор негайно зациклюватися, але не надають ніякої інформації про фактичне значення вихідних даних.

Атака з відомими вхідними даними: протягом цього виду атаки зловмисник може спостерігати частину вхідних даних, але не може маніпулювати ними. Знання вхідних даних може використовуватися для обмеження числа можливих вихідних значень для зменшення стійкості генератора або для підтримки інших атак, подібних, наприклад, атакам грубої сили. Атаки з відомими вхідними даними можливі, коли оцінка ентропії вхідних даних невірна, або якщо застосовано спостереження за пристроями введення. Перший випадок означає, що зібрана ентропія містить меншу ентропію щодо зловмисника, ніж користувач може допустити. Випадок спостережуваних вхідних даних може виникати, якщо вхідні дані видаленого користувача, які передавались через мережу, використовувалися як джерело вхідних даних. Загалом, будь-яка інформація, яка передається через мережу, є небезпечним джерелом випадковості [2].

Одним з прикладів атак з відомими вхідними даними є атака на генератор сеансових ключів Kerberos. Як ми вже показали, стійкість генератора складає як мінімум 32 біти. Випадкова функція переініціалізовується з 32-бітовим ключем кожного разу, коли запитується DES ключ.

Майже вся ентропія ключа міститься в 12 найменших значущих бітах початкового числа. Залишкові перші 20 бітів залишаються постійними протягом приблизно 12 днів (2^{20} секунд). Якщо початкове число визначається одного разу через 32 бітову атаку грубої сили, то ми можемо використовувати цю інформацію для зменшення стійкості генератора до 12 бітів. 12 бітів можуть бути легко зламані через грубу силу. 12 денний період достовірності перших 20 бітів робить сеансові ключі також вразливими для атаки попереднього обчислення. Зловмисникові доведеться обчислити шифротекст для заданого відкритого тексту і для всіх 2^{12} можливих ключів, якщо йому відомі перші 20 бітів. Згодом, весь шифротекст і ключові пари доведеться зберігати в сортованій таблиці.

Для знаходження вірного ключа повинен бути знайдений тільки відповідний вхід в таблиці. З попереднім обчисленням сеансовий ключ може бути знайдений за декілька сотень мілісекунд, тоді як без нього це займає декілька секунд. Тому, як показав аналіз, знання частини вхідних даних набагато легше допомагає знайти DES-ключ.

Атаки розповсюдження компрометації стану: в цій атаці припускається, що в заданий момент зловмисник вже знає частину внутрішнього стану генератора. Атака намагається розширити це знання до подальших моментів часу і до попередніх або майбутніх вихідних даних, відповідно. Така ситуація може виникати, якщо генеруючий процес був розгалужений або якщо ГВБ був запущений в незахищеному стані.

Другий випадок трапляється, коли генератор використовує фіксоване початкове значення (наприклад, всі нулі) і має повну довіру до обробки вхідних даних, або якщо генератор був переініціалізований з файлу, доступного зловмисникові. Взагалі, всі ГПВБ піддаються атакам розповсюдження компрометації стану, оскільки вони ніколи не обробляють ніяких вхідних даних. Проте і інші генератори також є вразливими до цих атак. Розглянемо цей факт на прикладі генератора випадкових бітів ANSI X9.1 [2].

Цей ГВБ застосовує секретний і фіксований ключ K потрійного DES. Вхідні дані генеруються за допомогою поточного часу, внутрішнього стану X_i, $i \geq 1$, і функції шифрування DES E_K як

$$T_i = E_K\left(\text{currenttimestamp}\right),$$

$$\text{output}[i] = E_K\left(T_i \oplus X_i\right), \text{ i}$$

$$X_{i+1} = E_K\left(T_i \oplus \text{output}[i]\right).$$

Операція $\oplus$ представляє побітовий XOR оператор. У всіх наступних атаках ми припускаємо, що зловмисник має здатність дізнатися секретний ключ K.

Атака постійної компрометації: ця атака означає, що генератор ніколи повністю не відновлюється з компрометованого стану. Зловмисник здатний визначити майбутні і навіть попередні вихідні значення [2].

Припустимо, що є можливість знайти ключ K генератора ANSI X9.17. Оскільки ключ ніколи не замінюється ніякими новими вхідними даними, інформація обробляється до тих пір, поки весь генератор, що включає ключ, не буде переініціалізований. Через якийсь час ми виявимо два послідовних вихідних значення ($\text{output}[i]$, $\text{output}[i+1]$). Припускаючи, що поточна тимчасова мітка містить тільки 10 невідомих бітів, які представляють реалістичне значення, існує 2^{10} вгадувань для кожного T_i і T_{i+1}. X_{i+1} може бути обчислено двома різними методами:

$$X_{i+1} = D_K\left(T_{i+1} \oplus \text{output}[i+1]\right) \text{ і}$$

$$X_{i+1} = E_K\left(T_i \oplus \text{output}[i]\right),$$

де D_K встановлюється для DES розшифрування.

Для кожного вгадування T_i, X_{i+1} обчислюється і зберігається в сортованій таблиці. Згодом обчислення відбувається для кожного T_{i+1}. Правильне значення X_{i+1} виникає як результат обох обчислень. Тому необхідно лише приблизно 2^{11} обчислень для виявлення поточного стану X_{i+1} генератора (2^{10} обчислень для визначення X_{i+1} з T_i і T_{i+1}, відповідно).

Атака зворотного відстеження: ця атака використовує скомпрометований стан для отримання інформації про попередні вихідні дані. За допомогою генератора ANSI X9.17 легко дізнатися як про майбутні вихідні дані, так і про попередні дані, начебто використовувався один і той же метод як в постійно скомпрометованій атаці [2].

Атака інтегративного вгадування: в цій атаці зловмисникові відомий тільки поточний стан S генератора в час t, і він може спостерігати подальші вихідні дані. На відміну від атаки постійної компрометації, йому достатньо тільки знання функції вихідних даних, а не самих вихідних даних. Така функція може бути шифруванням за допомогою генерованого ключа. Ітеративна атака вгадування використовує вгадані, але не відомі вхідні дані для визначення стану S в час $t+\xi$ [2].

У випадку генератора ANSI 9.17 достатньо легко застосувати цю атаку для $\xi = 1$. Припустимо, що відомий поточний стан генератора в час i, включаючи K, X_i і $\text{output}[i]$, і що відома функція $\text{output}[i+1]$. Використаємо попереднє припущення, що час містить тільки 10 бітів ентропії. Тоді нам необхідно як максимум 2^{10} можливих вхідних значень, і для прогнозування внутрішнього стану X_{i+1} необхідно порівняння результатів з функцією $\text{output}[i+1]$.

Атака зустрічі посередині: дана атака комбінує методи ітеративної атаки вгадування і атаки зворотного відстеження.

Знання стану S у час t і $t+2\xi$ використовується для визначення стану в час $t+\xi$ [2].

Припустимо, що генератор ANSI X9.17 виробляє послідовність з восьми послідовних ключів шифрування для шифрування відкритого тексту. Вихідні дані генератора невідомі, але можливо дізнатися стани X_i і X_{i+1} і зашифрований шифротекст, які використовують ключ, вироблений в час $t+4$. Припустимо, що кожна тимчасова мітка містить 10 бітів

ентропії. Атака зустрічі посередині дозволяє знайти ключ з набагато меншими витратами, ніж 2^{80} спроб. Тим же способом, який застосовується для атак постійної компрометації, обчислимо X_{i+4} через вгадування $T_{i+1,i+2,i+3,i+4}$ та $T_{i+5.i+6,i+7,i+8}$. Як бачимо, необхідно приблизно 2^{41} обчислень. Значення обчислень з обох сторін зберігаються в двох списках і порівнюються один з одним. Буде знайдено 2^{16} збігів. Кінцевий 2^{16} ключовий пошук виявляє правильний ключ для відстежуваного шифротексту.

Питання тимчасової ентропії: ентропія тимчасової мітки повинна бути оцінена дуже ретельно. Зокрема, коли генератор використовується для генерації послідовних вихідних даних за дуже короткий період часу. Якщо два ключа генеруються через послідовні виклики ГВБ, то відповідні тимчасові мітки відрізнятимуться тільки декількома бітами. Ця властивість дозволяє застосувати інший вид атак, але особливо ефективно, і навіть ефективніше, це робить атака зустрічі посередині [2].

Аналіз генераторів випадкових бітів на вразливість можливим атакам

Проаналізуємо вразливість існуючих генераторів на вразливість вищеописаним атакам.

В генераторі /dev/random використовуються два відокремлені пули: один для обробки вхідних даних, інший – для вироблення вихідних даних. Це розділення запобігає ітеративним атакам вгадування завдяки тому, що вхідні дані безпосередньо не впливають на вихідні дані. До того ж, /dev/random є частиною ядра. Тому, атаки з вибраними вхідними даними є ледве можливі, оскільки генератор використовує дані безпосередньо з системних подій. Кожного разу, коли генератор генерує вихідні дані, частина результату домішується назад в пул. Це допомагає запобігати атакам зі зворотним відстеженням.

Як показав аналіз, генератор Yarrow дозволяє уникати ітеративні атаки вгадування. Це досягається за рахунок того, що вхідні вибірки не мають проміжного впливу на вихідні дані шляхом розділення пулів і ключа. Швидкий пул дозволяє генератору швидко відновлюватися із скомпрометованого ключа, повільний пул запобігає вадам, викликаним переоцінкою вхідної ентропії.

Генератор BBS має доказову непередбачуваність наступного біта, якщо прості множники N невідомі. Тому захист засновується на нездатності факторизації великих цілих чисел на прості компоненти. Проте, як виявлено в процесі аналізу, BBS генератор є уразливим до всіх спеціальних атак на ГВБ. У випадку розгалуження процесу, дотриманого генератором, всі майбутні генеровані випадкові біти початкового і імітованого процесу є однаковими.

Генератор AES був розроблений і проаналізований для забезпечення високого захисту проти криптографічних атак. Проте можливою проблемою генератора AES в режимі лічильника може бути хронометражна атака на значення лічильника. Якщо вміст лічильника стане відомим зловмиснику, то стійкість цього режиму буде зменшена на 128 бітів лічильника. Але цій атаці можна запобігти шляхом використання лічильника, який завжди витрачає одну і ту ж кількість часу для інкрементації свого значення.

Висновки

Враховуючи все вищесказане, можна зробити висновок, що якість ГВБ є дуже важливим чинником, який треба враховувати при побудові ГВБ. Важливими є період псевдовипадкової послідовності, основа алфавіту, структурна скритність, ентропія джерела ключів, складність формування послідовності тощо.

В процесі аналізу установлено, що стійкість ГВБ проти криптоаналітичних атак є визначальною їх характеристикою.

На сьогоднішній день найбільшу загрозу для ГВБ представляють криптоаналітичні атаки, атаки, засновані на вхідних даних, і атаки розповсюдження компрометації стану.

Атака на ГВБ може вважатися успішною, якщо її складність не менше складності атаки типу «груба сила».

Проте деякі з цих атак можна легко запобігати через прості контрзаходи. Атаки компрометації стану можна уникати шляхом частої зміни повного стану генератора. А потужність атак, заснованих на вхідних даних, може бути зменшена шляхом використання різних вхідних джерел і через комбінування всіх зібраних даних за допомогою криптографічної геш-функції.

Література

1. Птицын Н. Приложение теории детерминированного хаоса в криптографии / Н. Птицын. – М., 2002. – 81 с.

2. Rock A. Pseudorandom Number Generators for Cryptographic Applications / A. Rock. – Salzburg, 2005. – P. 57–65.

3. Kang Ju-Sung. Security frameworks for pseudorandom number generators / Ju-Sung Kang. – Information Center for Mathematical Sciences, Volume 8, Number 1, 2005. – P. 1–11.

4. L'Ecuyer Pierre. Random Number Generation / Pierre L'Ecuyer. – Canada, 2005. – P. 15–19.

Надійшла до редакції 28.01.2010

Рецензент: д-р техн. наук О.В. Потій, Харківський національний університет радіоелектроніки, Харків.

АНАЛИЗ АТАК НА ГЕНЕРАТОРЫ СЛУЧАЙНЫХ БИТОВ

Н.В. Шапочка

Проводится анализ генераторов случайных битов (ГСБ). На математическом уровне приведено описание ГСБ. Определено, что важной характеристикой ГСБ является их устойчивость к криптоаналитическим атакам при оценке криптографического алгоритма. Рассмотрены возможные атаки на ГСБ, показано, как они работают, и какое влияние имеют на ГСБ. Приведена классификация атак на ГСБ. Сделан анализ некоторых ГСБ на уязвимость описанным атакам.

Ключевые слова: генератор случайных битов, случайность, энтропия, статистические свойства, криптоаналитические атаки.

ANALYSIS OF ATTACKS ON RANDOM BIT GENERATORS

N.V. Shapochka

The analyzing of RBG's is taking place below. Our RBG is mathematically described. It is defined, that the resistance to cryptographic attacks is important, when we assess the RBG. Possible attacks against RBG's are examined, their influence on RBG's and working schemes are shown. The classification of the attacks is described. The analysis of some RBG's on vulnerability to described attacks is made.

Key words: random bit generators, randomness, entropy, statistical properties, cryptanalytic attacks.

Шапочка Наталя Вікторівна – аспірантка кафедри безпеки інформаційних технологій Харківського національного університету радіоелектроніки, Харків, Україна, e-mail: natali_shap@mail.ru.