УДК 519.712

П. В. Селюх
Національний технічний університет України “Київський політехнічний інститут”,
кафедра математичних методів захисту інформації

ОЦІНКА ПОТУЖНОСТІ МНОЖИНИ МОДУЛІВ RSA, СТІЙКИХ ДО КРИПТОАНАЛІЗУ

Розглянуто аспекти практичної стійкості криптосистеми RSA, проаналізовано стандартизовані методи генерації параметрів криптосистеми, стійкість криптосистеми на практиці. Отримано кількісні оцінки множини стійких параметрів для застосування під час генерації параметрів криптосистеми. Висунуто гіпотезу про поліноміальну потужність множини надійних модулів RSA.

Ключові слова: сильні прості числа, кількість надійних параметрів RSA.

ESTIMATION OF CAPACITY OF RSA MODULE SET, STABLE FOR CRYPTOANALYSIS

This paper deals with aspects of practical security of RSA cryptosystem, analyzes standardized methods of generating cryptosystem parameters, cryptosystem security in practice. This work presents quantitative evaluation of set of secure parameters for crypto parameters generation. Also was proposed the hypothesis of polynomial capacity of secure RSA module set.

Key words: strong prime numbers, amount of secure RSA parameters.

Вступ

Сьогодні однією із найпоширеніших асиметричних криптосистем залишається криптосистема RSA. На її основі будуються криптографічні протоколи, що вирішують завдання шифрування, електронного цифрового підпису, розподілу ключової інформації. В основу стійкості схеми RSA покладено складну теоретико-числову задачу факторизації: знаходження простих дільників числа n великої розмірності.

Питання стійкості криптосистеми на практиці пов’язане із задачею коректного вибору параметрів криптосистеми. Нещодавнє дослідження групи вчених на чолі з Ар’єном Ленстра [1] підтверджує існування проблеми впровадження криптосистеми RSA, а саме генерації параметрів, що забезпечуватимуть стійкість системи на практиці. Із досліджених близько 11,5 млн. сертифікатів ключів RSA було виявлено понад 26000 вразливих ключів завдовжки 1024 біти та 10 ключів завдовжки 2048 бітів. Вразливістю в цьому випадку вважали можливість знайти розклад модуля криптосистеми на множники. Реалізували цю вразливість завдяки використанню спільних дільників модулів, причому лише незначну кількість таких спільних дільників було виявлено внаслідок перегенерації ключа для одного і того самого власника. Більшою мірою така ситуація зумовлена наявністю глобальної проблеми генерації “якісних” модулів, що будуються із великих простих чисел [2, 3], причому автори роблять висновок, що перехід до модулів більшого розміру не призводить до очікуваного зменшення кількості вразливих ключів. Додаткового дослідження потребують механізми генерації простих чисел, які використовуються для побудови модулів.

З огляду на вищесказане формулюється задача уточнити практичну стійкість застосовуваних криптосистем, в основу яких покладено схему RSA. Розглянемо питання кількості “стійких”

модулів RSA. Якщо виявиться, що таких модулів у межах визначеної розмірності існує обмежена кількість, причому ця кількість є поліномом від довжини модуля, то такий факт гостро ставить питання про стійкість криптосистеми RSA на практиці, а саме належність алгоритму криптоаналізу RSA (шляхом факторизації модуля) до класу P.

Вибір параметрів криптосистеми

Визначимо, що для побудови криптосистеми RSA необхідно обрати відкритий ключ, що складається з модуля *n* та експоненти *e*. Модуль криптосистеми є добутком непарних простих чисел $p_i, i = 1,2,..,u$, де $u \geq 2$, а відкрита експонента *e* – ціле число, що набуває значення від 3 до $n-1$ і задовольняє умову $НСД(e, \lambda(n)) = 1$, де $\lambda(n) = НСК(p_1 - 1,..., p_u - 1)$ та деякі додаткові умови [4]. Як закритий ключ вибрано додатне ціле *d* таке, що задовольняє умову $e \cdot d \equiv 1 (\mathrm{mod}\, \lambda(n))$. Розміром ключа RSA вважатимемо довжину (розмірність) модуля *n* у бітах. NIST обмежує множину розмірів ключа значеннями 1024, 2048 та 3072 біти. Крім того, накладаються обмеження на генерацію модуля *n* лише як добутку двох простих чисел $n = p \cdot q$ [5].

Для розгортання криптосистеми RSA спочатку обирають експоненту *e*, фіксоване або випадкове ціле $2^{16} < e < 2^{256}$. Для генерації модуля завдовжки 1024 як прості числа *p*, *q* не дозволяється вибирати випадкові прості, а лише такі, що задовольняють умови:

$p-1$ має простий дільник p_1,

$p+1$ має простий дільник p_2,

$q-1$ має простий дільник q_1,

$q+1$ має простий дільник q_2.

Числа p_1, p_2, q_1, q_2 називатимемо допоміжними простими, числа *p* і *q* – простими із додатковими умовами.

Для модулів завдовжки 2048 та 3072 біти дозволяється використовувати випадкові прості *p*, *q*, які будуються за допомогою ймовірнісних або конструктивних алгоритмів генерації простих чисел [4]. У табл. 1 наведено обмеження на довжини чисел при застосуванні різних алгоритмів генерації простих чисел. Функція len(*p*) – бітова довжина числа *p*.

Таблиця 1

Допустимі значення розмірності допоміжних чисел p_1, p_2, q_1, q_2

Розмір n	Мінімальна довжина допоміжних простих чисел	Максимальне значення len(p_1)+len(p_2)	
		p, *q* ймовірно прості	*p*, *q* доказово прості
1024	> 100 біт	< 496 біт	< 239 біт
2048	> 140 біт	< 1007 біт	< 494 біти
3072	> 170 біт	< 1518 біт	< 750 біт

Ймовірно просте число *p* – це таке число, що генерується за таким алгоритмом: визначається бажана довжина числа *p*; генерується довільне непарне випадкове число; простота числа перевіряється за допомогою ймовірнісного тесту. Найпоширенішим із застосовуваних є тест Мілера–Рабіна. На вхід алгоритму перевірки на простоту подається число *N*. Потім виконуються такі кроки алгоритму:

1. Обчислити *t* і *s* такі, що $N-1 = 2^t s$, *s* має бути непарним.
2. Вибрати випадкове число *b*, $2 \leq b \leq N-2$.
3. Обчислити $y = b^s \bmod N$.
4. Якщо $y \neq 1$ і $y \neq N-1$, то поки $i < t$ і $y \neq N-1$ починаючи з $i = 1$ виконувати:

4.1. $y = y^2 \bmod N$. Якщо $y = 1$, тоді Вихід "*N* – складене число", інакше $i = i + 1$.

5. Якщо $y \neq N - 1$, тоді Вихід "*N* – складене число".

6. Вихід "*N* – ймовірно просте число".

Під час побудови модуля існують такі вимоги щодо кількості ітерацій перевірки згенерованих чисел за допомогою ймовірнісного тесту, оскільки із кожною повторною ітерацією тесту ймовірність за просте число прийняти складене зменшується. Для рівня помилки 2^{-100} нижче наведено таблицю із значеннями кількості необхідних ітерацій.

Таблиця 2

Мінімальна кількість ітерацій тесту Мілера-Рабіна при генерації простих чисел, що використовуються для побудови RSA, рівень помилки 2^{-100}

Параметри	Тест Мілера-Рабіна
$p_1, p_2, q_1, q_2 > 100$ біт p і q: 512 біт	Для p_1, p_2, q_1, q_2: 38 Для p і q: 7
$p_1, p_2, q_1, q_2 > 140$ біт p і q: 1024 біт	Для p_1, p_2, q_1, q_2: 32 Для p і q: 4
$p_1, p_2, q_1, q_2 > 170$ біт p і q: 1536 біт	Для p_1, p_2, q_1, q_2: 27 Для p і q: 3

Зауважимо, що використання ймовірно простих чисел при впровадженні криптосистеми не тільки суттєво збільшує час побудови модуля, але у випадку отримання складеного числа може призвести до збоїв, які, своєю чергою, знижують стійкість до криптоаналізу. Сьогодні відомі ефективні конструктивні алгоритми. Використовуючи їх можна звести задачу побудови модуля від задачі тестування простоти до задачі генерації доказово простого числа. Розглянемо алгоритми Муарера і Шейва–Тейлора [6, 7]. Це так звані алгоритми "торнадо" – в першій ітерації алгоритму використовується невелике просте число, яке знаходиться за допомогою, наприклад, алгоритму пробних ділень. В наступних ітераціях у визначеному інтервалі, що гарантує збільшення розмірності числа, що будується, визначається доказово просте число, що є базисним для побудови інтервалу в наступній ітерації.

Алгоритм Маурера генерації доказово простого випадкового числа.

На вхід алгоритму подається ціле число k – кількість бітів шуканого простого числа. Алгоритм використовує такі параметри: L – границю пробних ділень; M – параметр, що гарантує (забезпечує) існування шуканого простого числа. Рекомендоване значення $M = 20$.

1. Якщо $2^k < L$, то згенерувати випадкове непарне *k*-бітне число *N* та перевірити простоту за допомогою пробних ділень. Повторювати, поки не буде отримано просте число *N*. Вихід.

2. Якщо $k \le 2M$, то $r = \frac{1}{2}$, інакше виконувати наступне, поки не буде виконано умову $k - rk > M$.

2.1. Вибрати випадково дійсне число *s*, $0 \le s \le 1$, $r = 2^{s-1}$

3. $k_1 = \lfloor rk \rfloor + 1$. Повторити кроки 1–2 для побудови k_1-бітного простого числа *q*.

4. Обчислити $t = 2^{k-1} / (2q)$.

5. Вибрати випадкове ціле R, $t < R \le 2t$. $N = 2Rq + 1$.

6. Вибрати ціле число a, $1 < a < N - 1$. Якщо $a^{N-1} (mod\ N) = 1$ і $НСД(a^{2R} - 1, N) = 1$, то Вихід "*N* – просте число".

7. Інакше повторити кроки 5-6.

Розглянемо також другий конструктивний алгоритм Шейва–Тейлора побудови доказово простого випадкового числа. На вхід алгоритму подається ціле число k – кількість бітів шуканого простого числа. Алгоритм використовує параметр L – границю пробних ділень.

1. Якщо $2^k < L$, то згенерувати випадкове непарне k-бітне число N та перевірити простоту за допомогою пробних ділень. Повторювати, поки не буде отримано просте число N. Вихід.

2. Якщо k непарне, то прийняти $k_1 = (k+3)/2$. Якщо парне, то $k_1 = k/2+1$. Рекурсивно проходити алгоритм із вхідним параметром k_1, щоб побудувати k_1-бітне просте число.

3. Вибрати випадкове ціле число x, $2^{k-1} \le x < 2^k$.

4. Нехай t – найменше ціле більше ніж $x/(2q)$.

5. Якщо $2tq+1 \ge 2^k$, тоді нехай t буде найменшим цілим, більшим ніж $2^{k-1}/(2q)$.

6. Прийняти $N = 2tq+1$.

7. Вибрати випадкове ціле a, $1 < a < N-1$, прийняти $x = a^{2t} \bmod N$. Якщо $x \ne 1$, $НСД(x-1, N) = 1$, $x^q = 1 \bmod N$, то N – побудоване просте число.

8. Інакше прийняти $t = t+1$ і повторити кроки 5–7.

Проте наведені вище алгоритми гарантують лише простоту згенерованих чисел. Додатковими вимогами до чисел, які є дільниками модуля RSA, є їх так звана властивість "максимізація складності спеціалізованих алгоритмів факторизації" [8]. Наприклад, для алгоритмів групи $p \pm 1$ ця властивість реалізується за наявності у числа n такого простого дільника p, що $p \pm 1$ є степенево-гладким для достатньо великої границі гладкості [4, 9], тобто $p \pm 1$ має великі прості дільники. Такий дільник p прийнято називати "сильним" простим числом. Отже, постає важливе питання про кількість та характер розподілу таких сильних простих чисел.

Розподіл та кількість сильних простих чисел

Просте число p називатимемо класичним сильно простим, якщо виконуються умови:

$$p = 1 (mod\ r),\ p = -1 (mod\ s),\ r = 1 (mod\ t), \quad (1)$$

де r, s, t – великі прості числа. Тобто числа p, r, s, t представляються у вигляді $p = 2jr+1$, $p = 2ks-1$, $r = 2lt+1$, і причому що менші числа j, k, l, то краще.

Зауважимо, що узагальненням поняття класичного простого числа є поняття сильних чисел [8], які максимізують складність всіх відомих алгоритмів факторизації.

Рівест спростовував міркування щодо необхідності використання саме сильних простих чисел як множників модулів RSA [10]. На прикладі алгоритму Ленстри він хотів показати, що вибір p не впливає на ефективність отримання результату факторизації, і що взагалі для одержання модулів RSA достатньо обирати випадкові прості числа. Пізніше його міркування були спростовані, оскільки деякі алгоритми, зокрема ECM (Elliptic Curve Method), ефективніше працюють, якщо число $p-1$ є гладким. Розглянемо метод Гордона побудови сильних простих чисел.

1. Будується випадкове просте число s, виходячи із заздалегідь вибраної для нього розмірності. Для цього можна вибрати псевдовипадкове число x бажаної розмірності і за допомогою пробних ділень лишити в інтервалі $[x, x+\log_2 x]$ числа, що не мають малих дільників. Серед чисел, що лишились, за допомогою тесту на простоту обрати просте число s.

2. Аналогічно побудувати просте число t.

3. За допомогою пробних ділень и тесту на простоту побудувати просте число $r = 2lt+1$, перебираючи l в інтервалі $[1, \log_2 t]$.

4. Обчислити $u = u(r,s) = (s^{r-1} - r^{s-1}) \bmod rs$.

5. Якщо u непарне, то приймаємо $p_0 = u$, інакше $p_0 = u + rs$.

6. Тестувати на простоту числа виду $p = p_0 + 2krs$ для $k = 0,1,2,...$

Накладемо умову застосування сильних простих чисел при побудові модуля криптосистеми. Умови, що накладаються на дільники модуля [11]:

$$p_1 \mid p-1,\ p_1 \ge n^{x_1},\ p_2 \mid p_1-1,\ p_2 \ge n^{x_2},\ p_3 \mid p+1,\ p_3 \ge n^{x_3},\ 0 < x_1, x_2, x_3 < 0.5$$
$$q_1 \mid q-1,\ q_1 \ge n^{x_4},\ q_2 \mid q_1-1,\ q_2 \ge n^{x_5},\ q_3 \mid q+1,\ q_3 \ge n^{x_6},\ 0 < x_4, x_5, x_6 < 0.5 \quad (2)$$

Через $p_i(n)$ позначимо *i*-й найбільший дільник числа *n*. Нехай $w_i(n,x) = \#\{t : 1 \le t \le n, p_i(t) \le n^x\}$ – кількість додатних цілих чисел менших або рівних *n*, для яких $p_i(n)$ не більший за n^x. Використовуючи результати Кнута і Траба Пардо отримуємо, що ймовірність того, що випадково обране число матиме найбільший простий дільник, більший за деяку межу, виражається формулою

$$P\{q_1 > n^x\} = \int_x^1 \frac{dt}{t} = -\ln x. \quad (3)$$

Використовуючи оцінки, отримані Адамаром і Валле–Пуссеном, отримуємо таку оцінку кількості класичних сильних простих чисел в інтервалі [*a, b*]:

$$\pi_s(a,b) = -\ln x_1 \cdot \ln x_2 \cdot \ln x_3 \cdot \int_a^b \frac{dt}{\ln t}. \quad (4)$$

Вважаючи події існування описаних дільників незалежними, оцінкове значення кількості модулів RSA, що мають класичні сильні прості дільники, визначається як

$$(\pi_s(a,b))^2. \quad (5)$$

Порівняємо отриману оцінку із існуючими оцінками кількості модулів RSA, які утворюються із випадкових простих чисел (без вимог до "сильних" простих чисел). Модуль RSA є так званим напівпростим числом, оскільки він має рівно два прості дільники. Число, прості дільники якого менші за певну межу *B*, називатимемо *B*-гладким. У роботах [12, 13] розглядається питання розподілу напівпростих та гладких чисел, оцінки отримані із міркувань справедливості гіпотези Рімана. Ймовірнісна функція $g(y)$ відповідає ймовірності того, що число y виявиться напівпростим. Наведемо апроксимацію функції за допомогою сумування ряду за значенням *p* простим числам:

$$g(y) \approx \sum_{p \le \sqrt{y}} \frac{1}{p(\ln y - \ln p)}. \quad (6)$$

Використовуючи формулу Мертенса (7) та результати теореми Абеля

$$\sum_{p<x} \frac{\ln p}{p} = \ln x + O(1), \quad (7)$$

отримано

$$g(y) \approx \frac{\ln\ln y - \beta}{\ln y}, \text{де } \beta = 1 + \ln\ln 2. \quad (8)$$

Позначивши через $\psi(x, y)$ кількість всіх *y*-гладких чисел, менших за *x*, отримано рекурентну формулу для обчислення

$$\psi(x, p_k) = \sum_{i=0}^{t_k} \psi\left(\frac{x}{p_k^i}, p_{k-1}\right),\ t_k = \left[\frac{\ln x}{\ln p_k}\right], k > 1, \quad (9)$$

де p_k – *k*-те просте число ($p_1 = 2$), $\psi(x,2) = \left\lfloor \frac{\ln x}{\ln 2} \right\rfloor + 1$.

Існують наближені оцінки для обчислення кількості гладких чисел, оскільки із ростом границі гладкості складність обчислень зростає експоненційно. Хільдбрант [14] отримав оцінку

$$\psi(x,y) = x \cdot \rho(u)\left\{1 + O\left(\frac{\ln(u+1)}{\ln y}\right)\right\}, \text{де } x = y^u. \quad (10)$$

Тут $\rho(u)$ – функція Дікмана-Де Брюіна, $u = \ln x / \ln y$, $\rho(u)$ задовольняє диференційне рівняння $u\rho'(u) + \rho(u-1) = 0$ при $u > 1$. Але така апроксимація справедлива лише за одночасного росту *x* і *y*, зберігаючи сталим відношення $\ln x / \ln y$. Існує наближена формула для обчислень при $y < (\ln x \ln\ln x)^{1/2}$

$$\psi(x,y) = \frac{1}{\pi(y)!}\prod\left(\frac{\ln x}{\ln p}\right)\left\{1 + O\left(\frac{y^2}{\ln x \ln y}\right)\right\}. \quad (11)$$

Але для оцінювання потужності множини гладких чисел із практично використовуваними границями гладкості це наближення не є застосовним.

Розглянуті оцінки розподілу напівпростих чисел дають уявлення про потужність всієї множини. Для розв'язання задачі вибору криптопараметрів RSA необхідно обмежувати цю множину вибором лише таких напівпростих, які не є гладкими для деякої межі гладкості *B* для максимізації складності розв'язання задачі факторизації таких напівпростих чисел. У визначених вище позначеннях цікавить наступна оцінка кількості напівпростих чисел в інтервалі $[2,T]$, що не є *B*-гладкими

$$K(T,B) = \int_2^T \frac{\ln\ln x - \beta}{\ln x} dx - \psi(T,B), \text{ де } \beta = 1 + \ln\ln 2. \quad (12)$$

З іншого боку, оцінити кількість *B*-гладких чисел в інтервалі $[2,T]$ можна за допомогою комбінаторних методів та теорії груп. Як границі гладкості є сенс вибирати прості числа. Тобто, зафіксувавши $B = p_k$ як *k*-те просте число, формулюємо задачу оцінювання кількості комбінацій $x = \prod_{i=1,k} p_i^{\alpha_i}$, причому $x < T$.

Зауважимо, що оцінки (4) та (12) є близькими за значеннями, але оцінка (4) отримана значно простішим шляхом та враховує умову вибору "сильних" простих чисел.

Загалом для застосовуваних RSA криптосистем справедливі такі твердження: для забезпечення стійкості криптосистеми на практиці необхідно висувати обмеження на прості числа, які використовуються для генерації модуля; множини виходів алгоритмів генерації таких простих чисел є обмеженими. Тому автор висуває гіпотезу, що існує поліноміальний за складністю алгоритм, який перераховує множину "стійких" RSA модулів. Наведені вище оцінки підкріплюють цю гіпотезу.

Висновки

Існуючі стандарти впровадження і використання асиметричних криптосистем накладають обмеження на генерацію криптопараметрів. Такий стан речей зумовлений сучасними досягненнями у вирішенні задачі факторизації чисел великої розмірності. Цікавим для оцінювання стійкості RSA на практиці є висвітлене питання потужності, а точніше, характеру зростання потужності множини надійних криптопараметрів, а саме напівпростих чисел, які мають додаткові обмеження на характер дільників.

1. Lenstra A.K., Hughes J.P., et al.. Ron was wrong, Whit is right //2012. Електронний доступ http://eprint.iacr.org/2012/064. 2. Brenstein D.J., Chang Y.-A., et al.. Factoring RSA keys from certified smart cards: Coppersmith in the wild. ePrint, Cryptology ePrint Archive. Report 2013/599, 2013. 3. Heninger N., Durumeric Z., et al.. Mining your Ps and Qs: detection of widespread weak keys in network devices. Usenix Security, 2012. 4. Мухачев В. А., Хорошко В. А. Методы практической криптографии. – К.: ООО "Полиграф-Консалтинг". 2005. – 215 с. 5. Digital signature standard. National Institute of Standards and Technology. Federal Information Processing Standards Publication, FIPS PUB 186-4-1. 6. Maurer U.M. Fast generation of secure RSA-moduli with almost maximal diversity. // Advances in Cryptology/ Spinger-Verlag Berlin, 1990. 7. Shawe-Taylor J. Generating string primes. Electronic letters (22)16, 1986, P.875–577. 8. Кудин А. М. Методы тестирования чисел на простоту и построения простых чисел // Безопасность информации. – 1996. – № 3. – С.23–32. 9. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. – М.: МЦНМО, 2003. – 328 с. 10. Rivest R. L., Silverman R. D. Are

“strong” primes needed for RSA? RSA Laboratories Seminar Series // Seminars Proceedings, 1999. 11. Байденко П. В., Кудін А. М. Ефективність застосування алгоритмів факторизації до модулів криптосистеми RSA // X Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених “Теоретичні і прикладні проблеми фізики, математики та інформатики” 19–20 квітня 2012 р. – Збірка тез доповідей учасників. – К., 2012. – С.253–254. 12. Ишмухаметов Ш.Т. Методы факторизации натуральных чисел. LAP LAMBERT Academic Publishing. – 2014. – 256 с. 13. Ишмухаметов Ш. Т., Шарифуллина Ф. Ф. О распределении полупростых чисел // Изв. вузов. матем. 2014. – №8. – С.53–59. 14. Hildebrand A. On the number of positive integers ≤ x and free of prime factors> y // Journal of Number Theory. v.22. Issue 3. 1986. – P.289–307.

УДК 004.451, 004. 492

Я. Я. Стефінко, А. З. Піскозуб
Національний університет “Львівська політехніка”,
кафедра безпеки інформаційних технологій,
кафедра захисту інформації

ВИКОРИСТАННЯ ВІДКРИТИХ ОПЕРАЦІЙНИХ СИСТЕМ ДЛЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ В НАВЧАЛЬНИХ ЦІЛЯХ

Наведено інформацію про методологію тестів на проникнення, його методів і способів реалізації. Проаналізовано сучасні безкоштовні та з відкритим вихідним кодом програми. Розглянуто приклад тестування на проникнення в академічній сфері в навчальних цілях на базі Kali Linux і Metasploitable 2 Linux. Ці перевірки і методи показують проактивні методи захисту та повинні допомогти поліпшити безпеку комп'ютерних систем і корпоративних мереж.

Ключові слова: Kali Linux, Metasploitable, пентест, проникнення, вразливість, Metasploit Framework, PTES, тест на проникнення, безпека, проактивний захист, корпоративні мережі.

USING KALI LINUX AND METASPLOITABLE FOR PENETRATION TESTING FOR STUDYING PURPOSES

This paper comprises information about penetration testing methodology, its methods and ways of implementation. The current free and open-source software has been analyzed. It has been done the example of penetration testing in academic field for studying purposes on the base of Kali Linux and Metasploitable2 Linux. This tests and techniques purpose proactive methods of defense and must help to improve security of computer systems and corporate networks.

Key words: Kali Linux, Metasploitable, pentest, penetration, vulnerability, Metasploit Framework, PTES, pentest, security, proactive defense, corporate networks.

Вступ

Сучасні комп’ютерні системи і мережі зазнають тисяч різних атак як ззовні, так і зсередини. Тому актуальним сьогодні є питання різнобічного підходу до питання захищеності. Саме тут виникає потреба оцінювання захищеності системи до зламу та запобігання його руйнівним наслідкам. Тести на проникнення є складовою повного аудиту безпеки.