توليد الاعداد شبه العشوائية

المقدمة

هناك العديد من الحالات في التشفير التي تتطلب القدرة على توليد ارقام عشوائية، مثلا:

1- عادة ما تكون مفاتيح التشفير الاساسية ومفاتيح الجلسات session keys قيم عشوائية.
2- تتضمن بروتوكولات ادارة المفاتيح وتوزيعها ارقام عشوائية.
3- يستخدم التشفير الانسيابي مولّد عشوائي لتوليد سلسلة ارقام عشوائية.

يتم استخدام مولّدات الارقام شبه العشوائية pseudo-random number generators (**PRNG**) لتوليد الارقام شبه العشوائية بدلا من كتابتها يدويا. تبتدأ المولّدات شبه العشوائية برقم عشوائي قصير يدعى البذرة seed ويعمل المولّد على توسيع تلك البذرة الى سلسلة من الارقام شبه العشوائية . عندما تختزل الارقام الناتجة الى ثنائيات فان المولّد يعرف بمولّد الثنائيات شبه العشوائية Pseude-random bit generator (**PRBG**).

يمكن تعريف **مولّد الثنائيات** شبه العشوائية بصورة شكلية كما يلي:

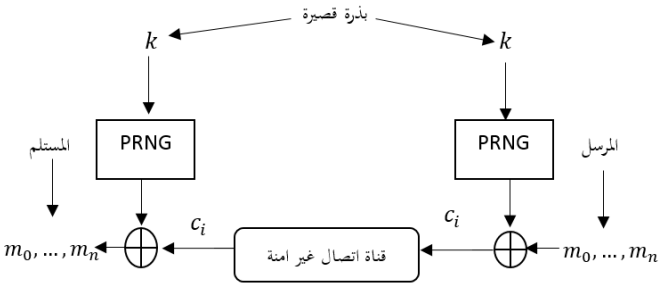
تعريف(9.1): ليكن لدينا عددين موجبين صحيحين k و ℓ بحيث $\ell \geq k + 1$. يكون مولّد الثنائيات هو الدالة

$$f: (\mathbb{Z}_2)^k \rightarrow (\mathbb{Z}_2)^\ell$$

ويمكن حسابها بوقت متعدد الحدود نسبة الى k. المدخل $s_0 \in (\mathbb{Z}_2)^k$ يدعى بالبذرة والاخراج $f(s_0) \in (\mathbb{Z}_2)^\ell$ هو السلسلة الثنائية المولّدة.

تكون الدالة $f(s_0)$ دالة محدّدة، بمعنى انها تعطي نفس المخرجات عند ثبوت البذرة. الهدف الاساسي للمولّد $f(s_0)$ هو عدم امكانية تمييز سلسلة الثنائيات الناتجة عن ثنائات عشوائية. عند تحقق هذا الشرط يكون المولّد "امنا".

ذكرنا في الفصل الاول مفهوم الامنية التامة. تعتبر طريقة تشفير one-time pad احد تطبيقات تلك الامنية، التي تتطلب ان يكون طول المفتاح العشوائي بطول الرسالة المطلوب تشفيرها، ويتولد النص المشفر بتطبيق عملية XOR بين ثنائيات النص الصريح وثنائيات المفتاح. يعتبر هذا الاسلوب امرا مربكا فيما يتعلق بالجانب العملي كونه يتطلب توليد ونقل مفتاح طويل جدا. لتخفيف هذه المشكلة، ذكرنا في حينه، انه يمكن استخدام مولّد اعداد شبه عشوائية لتوليد سلسلة ثنائيات المفتاح. حيث يتفق الطرفان Alice و Bob على بذرة صغيرة يتم تناقلها عبر قناة خاصة. يستخدم كلا الطرفان هذه البذرة لتوليد نفس سلسلة المفتاح، التي تستخدم للتشفير بتطبيق عملية XOR ايضا. هذا التشفير يدعى التشفير الانسيابي stream cipher وتعد البذرة هي المفتاح. يوضح الشكل (9.1) المبدأ العملي للتشفير الانسيابي باستخدام مولدات الاعداد شبه العشوائية.



شكل (9.1): المبدأ الاساسي لعمل التشفير الانسيابي.

فحوصات السلسلة شبه العشوائية. تستخدم الارقام شبه العشوائية في الكثير من تطبيقات الحاسوب كالمحاكاة، العينات، الفحوصات، وغيرها. يكفي في هذه التطبيقات ان تكون الارقام العشوائية ذات توزيع منتظم. هناك عدة مقاييس احصائية تستخدم لهذا الغرض، مثلا:

1. فحص التردد frequency test: يفحص عدد الاصفار والاحاد في السلسلة الناتجة ويشترط ان تكون مقاربة لسلسلة عشوائية.
2. فحص السلسلة run test: يفحص تكرار مقاطع معينة من الاصفار والاحاد وباطوال مختلفة، حيث يجب ان تكون مقاربة للسلاسل العشوائية.

يكون تحقيق امنية المولّدات شبه العشوائية في حالة التشفير اكثر صعوبة من بقية التطبيقات. فلايعني مجرد اجتياز اختبارات العشوائية ان يكون ذلك المولّد امنا. يرتبط مفهوم الامنية للمولّدات شبه العشوائية بما يعرف باقدرة على التنبؤ **predictability** والتي سنشير لها في الجزء التالي.

9.1 امنية المولّدات شبه العشوائية

يكون مولّد الثنائيات شبه العشوائية امنا اذا كانت السلسة التي طولها ℓمن الثنائيات الناتجة من المولّد تبدو كأنها عشوائة. هذا يعني انه من المستحيل تمييز هذه السلسلة عن سلسلة بطول ℓمن الثنائيات الناتجة بصورة عشوائية حقيقية. على سبيل المثال، عندما ينتج المولّد سلسلة فيها احاد بنسبة 2/3 فانه يمكن تمييزها بسهولة عن سلسلة عشوائية.

نناقش الان موضوع التمييز. افترض وجود سلسلة بطول ℓمن الثنائيات الناتجة بصورة عشوائية. يوجد هنالك 2^{ℓ} من السلاسل المحتملة، وعندما يتم اختيار احد

هذه السلاسل بصورة منتظمة عشوائيا، فان احتمالية اختيار احد تلك السلاسل هي $1/2^{\ell}$. وبذك فان احتمالية الخيوط العشوائية تتبع **توزيعا منتظما**.

افترض الان وجود مولّد ثنائيات شبه عشوائية f. افترض ايضا انه تم اختيار بذرة بطول k من الثانئيات عشوائيا، وان المولّد يستخدم هذه البذرة لتوليد سلسلة بطول ℓمن الثنائيات. طالما ان السلاسل النانجة تعتمد على قيمة البذرة فانه يوجد 2^{k} من السلاسل التي تظهر باحتمالية $1/2^{k}$ من اصل 2^{ℓ} من السلاسل المحتملة ، اما السلاسل $2^{\ell} - 2^{k}$ المتبقية فلا تحدث مطلقا. وبالتالي فان توزيع الارقام شبه العشوائية يكون **غير منتظم** للغاية.

على الرغم من كون التوزيعين العشوائي وشبه العشوائي مختلفان بصورة كبيرة، فان احتمالية تمييز سلسلتين من كل منهما تعتبر مهمة صعبة.(تحدث باحتمالية ضئيلة جدا).

التنبؤ بالثنائية التالية. يعتبر التنبؤ بالثنائية التالية next bit predictor من المفاهيم المرتبطة بامنية مولّدات الثنائيات، والتي تعمل كما يلي: ليكن لدينا f مولّد ثنائيات عشوائي بطول بذرة k وطول سلسلة ℓ. نعرّف دالة التنبؤ $\text{nbp}: (\mathbb{Z}_2)^{i-1} \to \mathbb{Z}_2$ لجميع قيم $i = 1, \ldots, \ell - 1$، حيث تعمل على اخذ اول $i - 1$ من الثنائيات النانجة من المولّدf (باستخدام بذرة عشوائية بطولk) وتحاول تنبؤ الثنائية رقم i.

نقول بان المولّد العشوائي f يكون امنا اذا كانت هناك احتمالية ضئيلة جدا negl ، بحيث يكون نجاح دالة التنبؤ nbp (بتنبؤ الثنائية i من الثنائيات $i - 1$) لايتجاوز $1/2 + \text{negl}$. يمكن استخدام دالة التنبؤ في تصمم مميّز distinguisher بين سلسلتين من الثنائيات. يوضح المنهج (9.1) كيفية تصميم المميز.

يكون ادخال المميز سلسلة بطول i من الثنائيات، ويقوم باستخدام الدالة nbp بتنبؤ الثانئية رقم i اعتمادا على سلسلة اول $i-1$ من الثانئيات. عندما تكون القيمة المتنبئة هي نفس القيمة الحقيقية للثنائية i، عندها يخرج المميّز القيمة 1، مما يعني ان سلسلة الثانائيات المدخلة ناتجة عن مولّد اعداد شبه عشوائية. والا فان اخراج المميز هو 0.

Construction (9.1): **Distinguisher**.

Input: sequence $(z_1, \dots, z_i)$ of bits.

Output: decision 1 or 0.

$z \leftarrow \text{nbp}(z_1, \dots, z_{i-1})$
if $z = z_i$ **return** 1
else return 0

9.2 تصميم المولّدات شبه العشوائية

هناك عدة طرق يمكن فيها تصميم مولّدات الاعداد شبه العشوائية. يمكن تلخيص تلك الطرق بالعناوين التالية:

1. طرق خاصة الغرض: تشمل الطرق التي صممت بصورة خاصة لتوليد اعداد شبه عشوائية.
2. دوال البعثرة شبه العشوائية: يتم استخدام دوال البعثة(التشفير الكتلي) في توليد اعداد شبه عشوائية وذلك لكون مخرجاتها غير قابلة للتمييز عن مخرجات دوال عشوائية صحيحة.
3. مناهج التشفير معلن المفتاح: تستخدم بعض مناهج التشفير معلنة المفتاح لتوليد الارقام العشوائية.
4. دوال النحت وشفرات توثيق الرسائل: وهذه الطرق هي المحبذة عمليا، حيث يمكن اعتبار SHA-1 كطريقة لتوليد ارقام عشوائية.

سوف نستعرض هذه الطرق بصورة مفصّلة.

9.2.1 المولّدات العشوائية خاصة الغرض.

1. مولّد التطابق الخطي Linear congruential generator

يعتبر هذا المولّد من اشهر المولّدات ولكنه غير امن. تعتمد فكرة هذا المولّد على استخدام بذرة ومن ثم توليد سلسلة من الاعداد من تلك البذرة. يتم توليد العدد باستخدام دالة خطية للعدد السابق. جميع العمليات تكون $\bmod M$. تكون مخرجات هذا المولّد هي الثنائيات الاولى من سلسلة الاعداد الناتجة. يوضح المنهج (9.2) طريقة عمل هذا المولّد.

Construction (9.2): Linear congruent generator

Input: integer $M \geq 2$, integers $a, b \in \{1, \ldots, M-1\}$, length of bit string ℓ, where $\ell \leq M$.

Output: bit sequence $< z_1, \ldots, z_\ell >$.

$k = 1 + \lfloor \log_2 M \rfloor$

Generate random seed s_0 of length k, where $s_0 \in \{1, \ldots, M-1\}$

for $i = 1$ to ℓ:

Compute $s_i = (as_{i-1} + b) \bmod M$

$z_i = s_i \bmod 2$

return $z_1, \ldots, z_\ell$

المثال التالي يوضح طريقة عمل مولّد التطابق الخطي.

مثال(9.1): افترض $a = 3, b = 5, M = 31, \ell = 10, \text{and } k = 5$ يقوم المولّد بتوليد الاعداد وفق العلاقة $s \rightarrow 3s + 5 \bmod 31$. لاحظ ان $13 \rightarrow 13$. عندما يتم اختيار اي بذرة ما عدا 13فان البذرة تمثل بداية الاعداد

وبقية الاعدد يتم توليدها. اخيرا، يتم استخلاص الثنائية الاولى(ذات الوزن الاقل) من كل عدد ناتج. فعندما تكون البذرة صفر سوف نحصل على الاعداد: 5,20,3, 14,16,22,9,1,8,29، والتي تنتج سلسلة الثنائيات: 1010001101 ، في حين تنتج البذرة 13 السلسلة الثائية: 1111111111.

Construction (9.3): BBS generator

Input: let k be length of seed, p, q be two $(k/2)$-bit primes such that $p = q = 3 \bmod 4$, length of bit generated bits ℓ.

Output: bit sequence $< z_1, \dots, z_\ell >$.

Define $n = pq$.

Generate random seed s_0 randomly from QR_n (we can select $s_0 \in \mathbb{Z}_n^*$, then set $s_0 = {s_0}^2 \bmod n$).

For $i = 0$ to $\ell - 1$:

- Compute $s_{i+1} = {s_i}^2 \bmod n$
- $z_i = s_i \bmod 2$

return $z_1, \dots, z_\ell$

2. مولّد Blum–Blum–Shub (BBS)

يعتبر مولّد BBS واحدا من مولّدات الثنائيات المشهورة. تعتمد فكرته على زمرة البقايا التربيعية **quadratic residue** للاعداد $\bmod n$، والتي تعرف لاي عدد صحيح فردي n بالشكل: $QR_n = \{x^2 \bmod n : x \in \mathbb{Z}_n^*\}$. يوضح عمل المولّد BBS في التصميم (11.3).

يعمل المولّد ببساطة باختيار قيمة $s_0 \in QR(n)$ ومن ثم توليد الاعداد $s_1, \dots, s_\ell$، بعدها يتم تقليص كل عدد $\bmod 2$ للحصول على الثنائيات. هذا

يعني ان الثنائيات تنتج وفق القانون التالي: $z_i = \left(s_0^{2^i} \bmod n\right) \bmod 2 : i = 1, \dots, \ell$.

i	s^i	z_i
0	20749	
1	143135	1
2	177671	1
3	97048	0
4	89992	0
5	174051	1
6	80649	1
7	45663	1
8	69442	0
9	186894	0

i	s^i	z_i
10	177046	0
11	137922	0
12	123175	1
13	8630	0
14	114386	0
15	14863	1
16	133015	1
17	106065	1
18	45870	0
19	13171	1
20	48060	0

شكل(9.2): ثنائيات المولد BBS.

نعطي الان مثالا لكيفية عمل المولّد BBS.

مثال(9.2): افترض ان $n = 192649 = 383 \times 503$ و$s_0 = 101355^2 \bmod n = 20749$. تظهر اول عشرون ثنائية من قبل المولّد BBS كما في شكل (9.2) وبالتالي فان سلسلة الثنائيات الناتجي هي: 11001110000100111010.

الاساس الرياضي للمولّد BBS. عندما تكون لدينا الزمرة $\mathbb{G}$، فان العدد $y \in \mathbb{G}$ يدعى بقية تربيعية quadratic residue اذا كان هناك عدد اخر $x \in \mathbb{G}$ بحيث ان $y = x^2$. الاعداد التي ليست بقايا تربيعية تعرف quadratic non-residue.

الزمرة $\mathbb{Z}_p^*$. في حالة الزمرة $\mathbb{Z}_p^*$ فان y هو بقية تربيعية اذا وجد $y = x^2 \bmod p$.
يرمز لزمرة البقايا التربيعية $\bmod p$ بالرمز QR_p ويرمز لزمرة البقايا غير التربيعية $\bmod p$ بالرمز QNR_p.

نعرف الان $\mathcal{J}_p(x)$، والذي يدعى برمز Jacobi للعدد $x \bmod p$، بالشكل التالي، بحيث $p > 2$ و $x \in \mathbb{Z}_p^*$:

$$\mathcal{J}_p(x) = \begin{cases} +1 \text{ if } x \in QR_p \\ -1 \text{ if } x \in QNR_p \end{cases}$$

فرضية(9.1): ليكن $p > 2$، فان: $\mathcal{J}_p(x) = x^{\frac{p-1}{2}} \bmod p$.

الشئ المفيد في فرضية (9.1) اعلاه هي انه من الممكن اختبار كون العدد $x \in \mathbb{Z}_p^*$ بانه بقية تربيعية ام لا عن طريق حساب $\mathcal{J}_p(x)$، فان كانت النتيجة 1 فهو بقية تربيعية، والا فهو بقية غير تربيعية.

الزمرة $\mathbb{Z}_n^*$. الان نفترض ان لدينا الزمرة $\mathbb{Z}_n^*$، حيث $n = pq$ حاصل ضرب عددين اوليين. يمكن تعريف زمرة البقايا التربيعية لهذه الزمرة بالشكل:

$$QR_n = \{x^2 \bmod n : x \in \mathbb{Z}_n^*\}.$$

والتي يعرّف لها الرمز Jacobi بالشكل: $\mathcal{J}_n(x) = \mathcal{J}_p(x).\mathcal{J}_q(x)$.

نعرّف الرمز $\mathcal{J}_n^{+1}$ للاشارة الى مجموعة عناصر $\mathbb{Z}_n^*$ التي لها Jacobi 1+، وبنفس الاسلوب نعرّف $\mathcal{J}_n^{-1}$.

اذا كان العدد $x \in \mathbb{Z}_n^*$ بقية تربيعية في الجموعة QR_n فأن $\mathcal{J}_n(x) = +1$.

على كل حال، يمكن ان يكون $\mathcal{J}_n(x) = +1$ عندما

$$\mathcal{J}_p(x) = \mathcal{J}_q(x) = -1،$$

بمعنى ان x هي ليست بقية تربيعية mod n. تسمى قيم x التي تحقق هذا الاختبار الخاطئ بكونها اعداد شبه تربيعية pseudo-square، ويمكن التعبير عن زمرة الاعداد شبه التربيعية بالرمز:

$$\widehat{QNR_n} = \{x \in \mathbb{Z}_n^* | \mathcal{J}_n(x) = +1 \text{ but } x \notin QR_n\}$$

فرضية (9.3): عندما تكون $n = pq$ فان:

1. نصف عناصر الزمرة $\mathbb{Z}_n^*$ بالضبط هي موجودة في $\mathcal{J}_n^{+1}$.
2. زمرة QR_n موجودة ضمن $\mathcal{J}_n^{+1}$.
3. نصف عناصر الزمرة $\mathcal{J}_n^{+1}$ بالضبط موجودة في QR_n والنصف الاخر موجود في $\widehat{QNR_n}$.

من الجدير بالذكر ان $|QR_n| = |\widehat{QR_n}| = (p-1)(q-1)/4$.

نستنتج مما سبق بأن اختبار كون $x \in QR_n$ يتطلب وجود عوامل n، حساب $\mathcal{J}_p(x)$و $\mathcal{J}_q(x)$ ومن ثم فحص كون $\mathcal{J}_p(x) = \mathcal{J}_q(x) = +1$.

امنية المولّد BBS. تعتمد امنية المولّد BBS على مسألة اختبار كون $x \in QR_n$، بحيث $n = pq$ هي حاصل ضرب عددين اوليين غير معلومين. تتطلب هذه المسألة التمييز بين كون العدد x بقية تربيعية ام بقية شبه تربيعية. وكما ذكرنا اعلاه، عندما يكون تحليل $n = pq$ ممكنا فانه يسهل حل المسألة وذلك بحساب $\mathcal{J}_p(x)$ و$\mathcal{J}_q(x)$. وبالتالي فان كان $\mathcal{J}_n(x) = +1$ فان العدد x بقية تربيعية اذا وفقط اذا كان $\mathcal{J}_p(x)$ ($\mathcal{J}_q(x)$) يساوي 1. نؤكد على انه هذه المسألة لايمكن حلها ما لم يتم تحليل قيمة n.

نشير الى ميزة مهمة تتعلق بامنية المولّد BBS. عندما تكون $n = pq$ بحيث $p = q = 3 \bmod 4$، فانه ينتج ان لجميع البقايا التربيعية x يوجد جذر وحيد فقط وهذه الجذر هو ايضا بقية تربيعية. وبالتالي فان جميع القيم التي نحصل عليها من المولّد BBS هي اعداد مبعثرة ضمن الزمرة QR_n.

مثال(9.3): عندما تكون $n = 253 = 11 \times 23$، فان

$$|QR_n| = \frac{(11-1)(22-1)}{4} = 55$$

Construction (9.4): RSA–based PRBG generator

Input: length of seed, k. p, q be two $(k/2)$-bit primes, length of bit generated bits ℓ, and b is an integer such that $\gcd(b, \emptyset(n)) = 1$.

Output: bit sequence $z =< z_1, \dots, z_\ell >$.

Define $n = pq$.

Generate random seed $s_0 \in \mathbb{Z}_n^*$ randomly.

for $i = 2$ to ℓ:

- Compute $s_{i+1} = s_i{}^b \bmod n$
- $z_i = s_i \bmod 2$

return $z_1, \dots, z_\ell$

9.2.2 المولّدات العشوائية باستخدام التشفير معلن المفتاح

يتم استخدام تشفير RSA كطريقة لتوليد اعداد شبه عشوائية امنة. يختار المولّد في البداية عنصر ينتمي للزمرة $\mathbb{Z}_N$ ثم يولّد سلسلة من الاعداد التي تنتمي لنفس الزمرة، حيث ان كل عنصر ينتج عن تطبيق تشفير RSA على العنصر الذي قبله. يتم اختيار الثنائيات الاولى (ذات الوزن الاقل) لتمثل سلسلة الثنائيات الناتجة. يوضح المنهج (9.4) منهج PRBG باعتماد تشفير RSA.

مثال(9.4): افترض ان $n = 91261 = 263 \times 347$ ، $b = 1547$، و $s_0 = 75634$. اول عشرون ثنائية يولدها مولّد RSA يوضحها الشكل (9.3).

i	s^i	z_i
0	75634	
1	31483	1
2	31438	0
3	51968	0
4	39796	0
5	28716	0
6	14089	1
7	5923	1
8	44891	1
9	62284	0

i	s^i	z_i
10	11889	1
11	43467	1
12	71215	1
13	10401	1
14	77444	0
15	56794	0
16	78147	1
17	72137	1
18	89592	0
19	29022	0
20	13356	0

شكل (9.3): نتائج مولد RSA.

الثنائيات الناتجة عن هذه البذرة هي: 10000111011110011000.

9.2.3 مولّدات الاعداد العشوائية باستخدام دوال البعثرة العشوائية.

احد الطرق المشهورة لتصميم مولّدات الارقام شبه العشوائية هي باستخدام دوال البعثرة شبه العشوائية كمكون اساسي في تصميم تلك المولّدات. لاي كتلة من البيانات الصريحة، تنتج دوال البعثرة شبه العشوائية (التشفير الكتلي)كتلة شبه عشوائية. بحيث لايمكن ايجاد ترابط بين مدخلاتها ومخرجاتها.

عند استخدام دوال بعثرة شبه عشوائية مبرهنة الامنية في تصميم مولّدات الاعداد شبه العشوائية، فانه يمكن برهنة امنية تلك المولّدات بسهولة.

هناك نمطان لاستخدام التشفير الكتلي (دوال البعثرة شبه العشوائية) لبناء PRNG : نمط CTR ونمط OFB. يوضح المنهج (9.5) النمط الاول، اما منهج (9.6) يوضح المنهج الثاني. في كل منهج، تتكون البذرة $s =< k, c >$ من جزئين: مفتاح التشفيرk وقيمة c الناتجة بعد توليد كل عدد عشوائي جديد. على سبيل المثال، عند استخدام AES-128 فان طولي المفتاح وقيمة c سيكونان 128 ثنائية. تزداد قيمة c في نمط CTR بمقدار 1 بعد توليد كل عدد جديد. في حين تأخذ قيمة c ناتج التشفير بعد توليد كل عدد جديد. يشترك كلا النمطين بتوليد كتلة ثنائيات في كل مرة.

Construction (9.5): CTR–based PRBG generator

Input: pseudo permutation function f, block length b, encryption key k, length of bit generated bits ℓ, initial c value.

Output: bit sequence $z =< z_1, \dots, z_\ell >$

$z = []$

while $|z| < \ell$

- $c = c + 1 \bmod 2^b$
- $z = z || Enc_k(c)$

return $z_1, \dots, z_\ell$

Construction (9.6): OFB–based PRBG generator

Input: pseudo permutation functionf, block length b, encryption key k, length of bit generated bits ℓ, initial c value.

Output: bit sequence $z =< z_1, \dots, z_\ell >$.

$z = []$

while $|z| < \ell$

- $c = Enc_k(c)$
- $z = z||c$

return $z_1, \dots, z_\ell$

9.2.4 خوارزمية RC4

طوّرت خوارزمية RC4 للتشفير الانسيابي من قبل Ron Rivest عام 1987 وتستخدم كثيرا في بروتوكولات الويب (مثلا SSL/TLS) و بروتوكولات الشبكات اللاسلكية (802.11b WEP). صُمّمت هذه الخوارزمية للتعامل مع قيم بطول 8 ثنائية. على الرغم من استخدام RC4 الشائع فانه ثبت احتوائها على الكثير من

Algorithm (9.1): Setup algorithm.

Input: string s of bytes.

Output: array S.

for $i = 0:255$

$S[i] = i$

$j = 0$

for $i = 0:255$

$k = s[i \ mod \ |s|]$

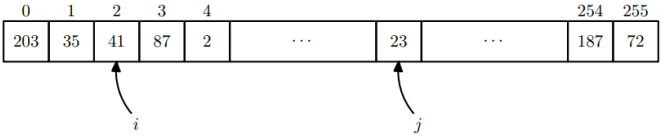
$j = (j + S[i] + k) mod \ 256$

swap $(S[i], S[j])$

return S

الثغرات الامنية، لذا ينصح بعدم استخدامها في المشاريع الجديدة.

يشتمل قلب هذه الخوارزمية على مولّد ارقام شبه عشوائية يدعى RC4 PRNG. يتضمن هذا المولّد حالة داخلية بشكل مصفوفة S مكوّنة من 256 بايت وتستخدم ايضا مؤشرين i, j، يستخدمان للاشارة الى قيم تلك المصفوفة. تتضمن المصفوفة S جميع الاعداد 255...0 بصورة مبعثرة وبدون تكرار. يوضح الشكل (9.4) مثال لحالة RC4.



شكل(9.4): مثال لحالة RC4.

تستخدم خوارزمية التشفير الانسيابي RC4 المفتاح S كبذرة للمولد العشوائي

Algorithm (9.2): Stream generator.

Input: array S.

Output: Pseudo-random numbers

$i = 0, j = 0$

repeat

$i = (i + 1) mod\ 256$

$j = (j + S[i]) mod\ 256$

swap $(S[i], S[j])$

return $S[S[i] + S[j] mod\ 256]$

ويستخدم لانشاء الحالة الابتدائية للمصفوفة S بشكل بعثرة شبه عشوائية. توضح الخوارزمية (9.1) كيفية الاعداد لخوارزمية RC4.

بعد تهيئة المصفوفة S يتم توليد الاعداد العشوائية وبواقع بايت واحد في كل مرة، كما هو موضح في خوارزمية (9.2).

تتميز خوارزمية RC4 بسرعة تنفيذها برامجيا اسرع من تنفيذها ماديا. اما من حيث الامنية فقد ثبت مؤخرا تعرضها لبعض المهاجمات حيث تبين ان مخرجات الارقام العشوائية تنحاز لبعض القيم وبالتالي يمكن تمييزها عن الارقام العشوائية.

تمارين

1. افترض ان لديك مولّد التطابق الخطي والمعرّف بالشكل $\mathrm{s_i} = (\mathrm{as_{i-1}} + \mathrm{b}) \bmod \mathrm{M}$ حيث جميع العمليات تكون في $\mathbb{Z}_M$ وافترض ان $a \neq 1$.
   أ. برهن ان $s_i = s_0 a^i + \frac{b(a^i-1)}{a-1}$ لجميع القيم $i \geq 0$.
   ب. تعرّف "دورة" المولّد بانها اصغر عدد صحيح موجب t بحيث $\forall\, i \geq 0 : z_{i+t} = z_i$. برهن ان $t = 1$ اذا كان $s_0 = b/(a-1)$.
2. افترض مولّد RSA بحيث $n = 36863$، $b = 229$، والبذرة $s_0 = 25$. احسب اول 100 ثنائية لهذا المولّد.
3. ليكن لدينا المولّد اللوغارتمي المتقطع والمعرّف بالشكل $x_i = g^{x_{i-1}} \bmod p$، بحيث p هو عدد اولي بطول k من الثانيات، g هو مولّد الزمرة، والبذرة x_0 هي اي عنصر في $\mathbb{Z}_p^*$، تولّد الثنائيات $i \geq 0$ بالشكل التالي

$$z_i = \begin{cases} 1 \ if \ x_i > p/2 \\ 0 \ if \ x_i < p/2 \end{cases}$$

افترض $p = 21383$، المولّد $g = 5$، والبذرة $s_0 = 15886$.
احسب اول 100 ثانئية يتم انتاجها من قبل هذا المولّد.

4. ماهو المفتاح الذي لايغير ترتيب المصوفة الابتدائية S في خوارزمية $RC4$، بمعنى ان تكون القيم 255...0 بدون بعثرة.

5. افترض ان لديك طرفين مرسل ومستلم يرغبان بالتواصل السري باستخدام مفتاح تشفير k بطول 128 ثنائية. يقوم المرسل بتشفير الرسالة m بالمنهج التالي:

أ. كيف يقوم المستلم بفك شفرة الرسالة واسترجاع m.

1. Choose a random 80-bit v.
2. Compute the ciphertext $c = RC4(v||k) \oplus m$
3. send $(v||c)$

ب. عندما يلاحظ الخصم الازواج $(v_1||c_1)$،$(v_1||c_1)$ ، ... المرسلة بين المرسل والمستلم. كيف يستطيع الخصم ان يعرف ان هذه الرسائل مشفرة بنفس المفتاح؟

ت. كم هو عدد الرسائل التي ترسل قبل ان تتكرر قيم المفتاح؟ استخدم مهاجمة يوم الميلاد.

6. ليكن لدينا مولدين من نوع التطابق الخطي

$$S_{i+1} = (6S_i) mod\ 13$$
$$S_{i+1} = (7S_i) mod\ 13$$

اكتب السلاسل الناتجة من هذين السلسلتين وبين اي واحد منهما يبدو عشوائيا.