دراسة وتصميم خوارزمية تشفير للبيانات المنقولة عبرالشبكة[١]

المهندس ناهي يوسف الشاهين[٢] الأستاذ الدكتور المهندس نديم شاهين[٣]

الملخص

في هذا البحث تم تصميم برنامج خوارزمية تشفير وتنفيذه للبيانات المنقولة عبر شبكة حاسوبية حيث كُتب البرنامج بلغة C. وصممت هذه الخوارزمية لتكون معقدة وذات سرية مرتفعة، حيث استخدمت خليطاً من خوارزميات التشفير المتناظرة وغير المتناظرة بالإضافة إلى وضع خوارزمية خاصة لتوليد مفتاح التشفير توليداً معقداً وعشوائياً وهذا ما جعل أية بيانات منقولة عبر الشبكة وكذلك مفاتيح التشفير المتبادلة بين أطراف الشبكة عصية بشكل كبير جداً على أية محاولة لكشفها ومعرفة محتواها. وهكذا نكون قد أمنّا السرية والأمن للبيانات المنقولة عبر أي شبكة تستخدم هذه الخوارزمية.

[١] أعد هذا البحث في سياق رسالة الماجستير للمهندس ناهي يوسف الشاهين بإشراف الأستاذ الدكتور المنهدس نديم شاهين.

[٢] كلية الهندسة الميكانيكية والكهربائية، جامعة دمشق.

[٣] كلية الهندسة الميكانيكية والكهربائية، جامعة دمشق.

١ - مقدمة introduction:

التشفير(Encryption):هو عملية تحويل المعلومات بطريقة ما إلى رموز سرية بحيث تصبح محمية من عمليات الوصول غير المرخص بها، أي أن معلومات الرسالة الواضحة المتناسقة M يتم تحويلها إلى رموز مبعثرة لا معنى لها C و هذا التحويل يتم عن طريق خوارزمية التشفير E انطلاقاً من مفتاح تشفير خاص بعملية التشفير K_e وفق العلاقة: $C = E\,K_e\,(M)$

وعكس هذه العملية يسمى فك التشفير حيث يتم استرجاع الرسالة الأصلية M باستخدام خوارزمية فك التشفير D انطلاقاً من مفتاح فك التشفير K_d وفق العلاقة: $M=D\,K_d\,(C)$

وهناك طريقتان رئيسيتان لتشفير المعلومات وهما:

١- طريقة التشفير المتناظرة (symmetric encryption): وفيها يكون مفتاح التشفير متماثلاًفي كلا الطرفين وخوارزمية التشفير مشابهة لخوارزمية فك التشفير. $(E = D\ ,\ K_e = K_d)$

٢- طريقة التشفير غير المتناظرة (asymmetric encryption):

وفيها يكون$(E \neq D\ ,\ K_e \neq K_d)$ وتسمى هذه الطريقة التشفير بالمفتاح العمومي.

(Public key Algorithm) حيث يسمى مفتاح التشفير K_e بالمفتاح العام Public key والمفتاح K_d بالمفتاح الخاص(Privet key).

أما الخوارزمية المعتمدة في هذه الدراسة فقد جمعت بين طريقتي التشفير معاً وبذلك حصلنا على خوارزمية تشفير معقدة وذات مستوى سرية مرتفع تؤمن حماية البيانات المنقولة بالإضافة إلى حماية مفاتيح التشفير الموزعة والمتبادلة بين عقد الشبكة.

٢ - المواصفات الرئيسة للخوارزمية المنفذة

تحتوي خوارزمية التشفير المنفذة ثلاث خوارزميات تشفير جزئية وهي:

١ – خوارزمية RSA

٢ – خوارزمية تشفير المعطيات الدولية IDEA

٣ – خوارزمية توليد المفتاح

٢-١ – خوارزمية RSA The RSA Algorithm

هي خوارزمية مفتاح عمومي يتلخص مبدأ عملها كما يأتي:

أولاً- نقسم الرسالة إلى كتل، بحيث يكون لكل كتلة قيمة ثنائية أصغر من عدد n ويكون التشفير وفك التشفير وفق الصيغة الآتية:

$C = M^e \bmod n$ النص المشفر

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$ النص غير المشَفر

حيث M هو الرسالة أو حرف من الرسالة أو بشكل أدق هو الرقم المسند إلى رمز أو إلى حرف من الرسالة. و تكون المفاتيح على النحو الآتي:

المفتاح العمومي: $Ku = \{e , n\}$

المفتاح الخصوصي: $Kr = \{d , n\}$

ونحصل على هذه المفاتيح بالشكل الآتي:

١ – نختار عدداً فردياً e وهو جزء من المفتاح العمومي

٢ – نختار عددين أوليين q,p بحيث يكون العدد $(p-١)(q-١)-١$ يقبل القسمة على e

٣ – نحسب n حيث $n = p.q$

حيث n كما ذكرنا الجزء الأول من كلا المفتاحين.

٤ – نحسب d حيث $d = (p-١)(q-١)(e-١) +١ / e$

وهو الجزء الثاني من المفتاح الخصوصي.

ونوضح هذه الخوارزمية بالمثال الآتي:

- نختار العدد الفردي $e = ٣$

- نختار عددين أوليين $p = ٥$ $q = ١١$ حيث:

$(p - ١)(q - ١) – ١ = ٣٩$ يقبل القسمة على e

- نحسب n : $n = q . p = ٥٥$

- نحسب d : $d = (p-١)(q-١)(e-١) + ١ / e = ٢٧$

فيكون لدينا $Kr = \{٢٧ , ٥٥\}$ $Ku = \{٣ , ٥٥\}$

بفرض لدينا $M = ٧$ عندها يكون:

التشفير $C = M^e \bmod n = ٧^٣ \bmod ٥٥ = ١٣$

فك التشفير $M = C^d \bmod n = ١٣^{٢٧} \bmod ٥٥ = ٧$

ومن أجل زيادة السرية والتخلص من مشكلة توزيع المفاتيح الخاصة بهذه الخوارزمية فقد جعل لكل عقدة من عقد الشبكة مفتاح خاص بها تقوم بتوليده بشكل سري، ومن ثَم على كل عقدة نشر المفتاح العام و الاحتفاظ بالمفتاح الخاص وبناء عليه ليس هناك من ضرورة لابتكار أية طريقة سرية لتبادل هذه المفاتيح وتقع مسؤولية تـأمين المفتـاح الخاص على كل عقدة بمعزل عن العقد الأخرى. و يتم الاتفاق بين عقد الشبكة علـى استراتيجية خاصة من أجل تغيير المفاتيح والتي يمكن أن تكـون دوريـة أو عنـدما تقتضي الحاجة وذلك من أجل رفع مستوى سرية طريقة التشفير المعتمدة.

٢-٢- خوارزمية International Data Encryption Algorithm IDEA

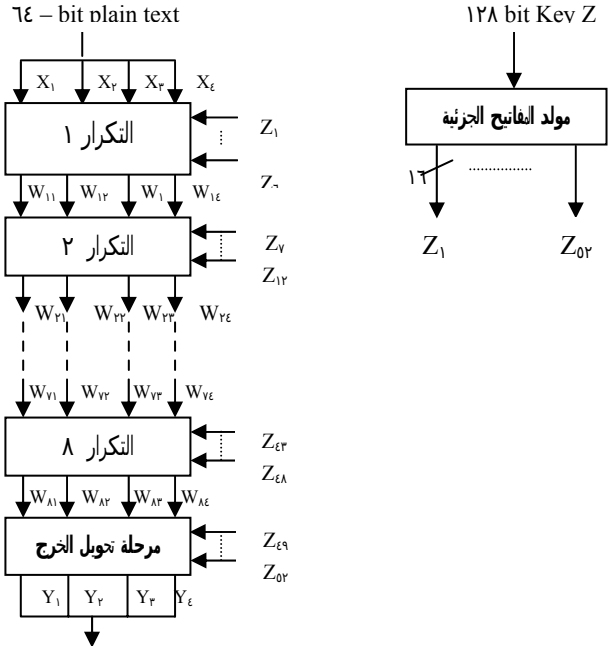
IDEA هي خوارزمية تشفير تقليدية كتلية التوجه Block–Oriented تقوم بتشفير كتل من الرسائل بطول ٦٤ bit باستخدام مفتاح بطول ١٢٨ bit.

تعمل هذه الخوارزمية على مبدأ أن يؤثر كل بت في النص الأصلي و كل بـت فـي المفتاح في كل بت في النص المشفر. يتحقق ذلك عبر استخدام ثلاث عمليات مختلفة. تأخذ كل عملية دخلين من ١٦ bitوتنتج خرجاً واحداً من ١٦ bit. وهذه العمليات هي:

- عملية XOR bit to bit. وترمز $\oplus$
- عملية جمع وأخذ باقي قسمة المجموع على$٢^{١٦}$. ($A+B \bmod ٢^{١٦}$) وترمز $\boxplus$

- عملية جداء وأخذ باقي قسمة الجداء على $2^{16}+1$. ($A*B \bmod 2^{16}+1$) ويعد الصفر مساوياً إلى 2^{16} ورمز ⊙ .

يوضح الشكل (٢) مخطط خوارزمية التشفيرIDEA. حيث نلاحظ دخلين هما النص المقروء ٦٤ bit والمفتاح ١٢٨bit. الشكل يبين أن الخوارزمية مؤلفة من ثماني مراحل أو تكرارات متبوعة بتحويل نهائي. نستخدم في كل مرحلة ستة مفاتيح جزئية ١٦ bit ما عدا المرحلة الأخيرة التي تستخدم أربعة مفاتيح ليكون مجموع المفاتيح الجزئية ٥٢ مفتاحاً. يبين الجزء الأيمن من الشكل (٢)عملية توليد المفاتيح الجزئية.



شكل (٢)

١-٢-٢ : دراسة مرحلة واحدة: Details of a single iteration

يمثل الشكل(٣) الخوارزمية المتكررة في جميع المراحل(مع دخول مختلفـة ومفـاتيح جزئية مختلفة طبعاً). ويكون الخرج بالنسبة لمرحلة واحدة على الشكل الآتي:

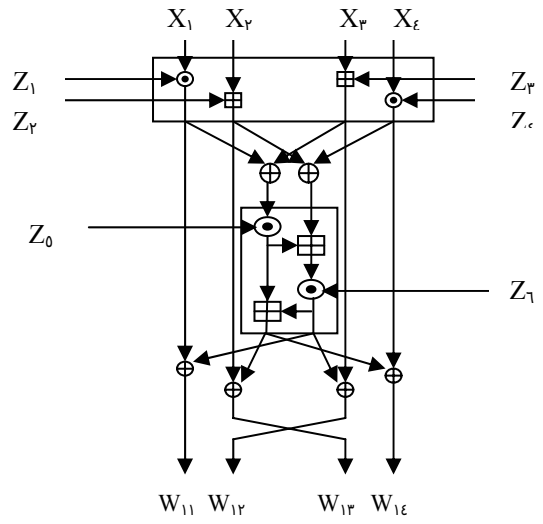
$$W_{11}=\{z_6 \odot [[z_0 \odot [(z_1 \odot x_1) \oplus (z_3 \boxplus x_3)] \boxplus [(z_2 \boxplus x_2) \oplus (z_4 \odot x_4)]]\} \oplus \{(z_1 \boxplus x_1)\}$$

$$W_{12}=\{z_6 \odot [[z_0 \odot [(z_1 \odot x_1) \oplus (z_3 \boxplus x_3)] \boxplus [(z_2 \boxplus x_2) \oplus (z_4 \odot x_4)]]\} \oplus \{(z_3 \boxplus x_3)\}$$

$$W_{13}=\{[z_6 \odot [[z_0 \odot [(z_1 \odot x_1) \oplus (z_3 \boxplus x_3)] \boxplus [(z_2 \boxplus x_2) \oplus (z_4 \odot x_4)]]] \boxplus [z_0 \odot [(z_1 \odot x_1) \oplus (z_3 \boxplus x_3)]]\} \oplus \{(z_2 \boxplus x_2)\}$$

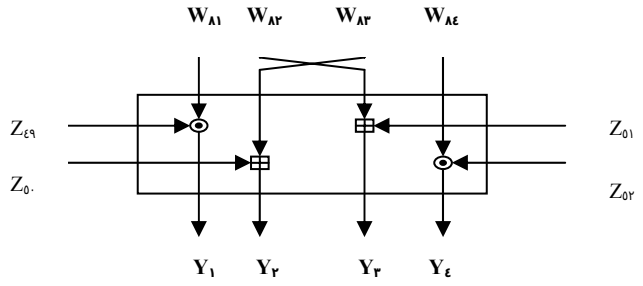
$$W_{14}=\{[z_6 \odot [[z_0 \odot [(z_1 \odot x_1) \oplus (z_3 \boxplus x_3)] \boxplus [(z_2 \boxplus x_2) \oplus (z_4 \odot x_4)]]] \boxplus [z_0 \odot [(z_1 \odot x_1) \oplus (z_3 \boxplus x_3)]]\} \oplus \{(z_4 \odot x_4)\}$$

وهذا يؤكد أن كل بت من الخرج تعتمد على كل بتات الدخل وعلى كل بتات المفـاتيح الجزئية.



شكل (٣)

ويبين الشكل(٤)المرحلة التاسعة من الخوارزمية و التي تدعى مرحلة التحويل النهائي.



شكل (٤)

٢-٢-٢ :توليد المفاتيح الجزئية في IDEA: Subkey Generation

تؤخذ المفاتيح الثمانية الأولى ($Z_1,\ldots,Z_8$) مباشرة من المفتاح الأساسي

(Z_1 هي ١٦bit الأولى ... وهكذا)، ثم نجري عملية إزاحة دورانية bit ٢٥ للمفتاح الأساسي و تؤخذ المفاتيح الثمانية الأخرى ($Z_9,\ldots,Z_{16}$)، ونتابع عملية الإزاحة الدورانية حتى توليد جميع المفاتيح الجزئية (٥٢مفتاحاً).

٢-٢-٣ :فك التشفير في IDEA IDEA Decryption

إن آلية فك التشفير مماثلة لآلية التشفير من حيث الجوهر. ولكن مع اختيار مختلف لمفاتيح فك التشفير الجزئية $u_1,\ldots,u_{52}$ التي يتم استخلاصها من مفاتيح التشفير، و الجدول الآتي يبين العلاقة بين مفاتيح التشفير ومفاتيح فك التشفير الجزئية إذْ:

$$z_i \odot z_i^{-1} = 1 \qquad -z_j \boxplus z_j = 0$$

فك التشفير		التشفير		
الموافقة لـ	المفاتيح الجزئية	أرقام الخانات الموافقة من المفتاح	أرقام المفاتيح الجزئية	المرحلة
$Z_{٤٩}^{-١}$ -$Z_{٢٠}$ -$Z_{٥١}Z_{٥٢}^{-١}$ $Z_{٤٧}Z_{٤٨}$	$U_١U_٢U_٣U_٤U_٥U_٦$	Z[١..٩٦]	$Z_١Z_٢Z_٣Z_٤Z_٥Z_٦$	التكرار ١
$Z_{٤٣}^{-١}$ -$Z_{٤٥}$ -$Z_{٤٤}Z_{٤٦}^{-١}$ $Z_{٤١}Z_{٤٢}$	$U_٧U_٨U_٩U_{١٠}U_{١١}U_{١٢}$	Z[٩٧..١٢٨;٢٦..٨٩]	$Z_٧Z_٨Z_٩Z_{١٠}Z_{١١}Z_{١٢}$	التكرار ٢
$Z_{٣٧}$-١ -$Z_{٣٩}$ -$Z_{٣٨}Z_{٤٠}^{-١}$ $Z_{٣٥}Z_{٣٦}$	$U_{١٣}U_{١٤}U_{١٥}U_{١٦}U_{١٧}U_{١٨}$	Z[٩٠..١٢٨;١..٢٥;٥١..٨٢]	$Z_{١٣}Z_{١٤}Z_{١٥}Z_{١٦}Z_{١٧}Z_{١٨}$	التكرار ٣
$Z_{٣١}$-١ -$Z_{٣٣}$ -$Z_{٣٢}Z_{٣٤}^{-١}$ $Z_{٢٩}Z_{٣٠}$	$U_{١٩}U_{٢٠}U_{٢١}U_{٢٢}U_{٢٣}U_{٢٤}$	Z[٨٣..١٢٨;١..٥٠]	$Z_{١٩}Z_{٢٠}Z_{٢١}Z_{٢٢}Z_{٢٣}Z_{٢٤}$	التكرار ٤
$Z_{٢٥}$-١ -$Z_{٢٧}$ -$Z_{٢٦}Z_{٢٨}^{-١}$ $Z_{٢٣}Z_{٢٤}$	$U_{٢٥}U_{٢٦}U_{٢٧}U_{٢٨}U_{٢٩}U_{٣٠}$	Z[٧٦..١٢٨;١..٤٣]	$Z_{٢٥}Z_{٢٦}Z_{٢٧}Z_{٢٨}Z_{٢٩}Z_{٣٠}$	التكرار ٥
$Z_{١٩}^{-١}$ -$Z_{٢١}$ -$Z_{٢٠}Z_{٢٢}^{-١}$ $Z_{١٧}Z_{١٨}$	$U_{٣١}U_{٣٢}U_{٣٣}U_{٣٤}U_{٣٥}U_{٣٦}$	Z[٤٤..٧٥;١٠١..١٢٨;١..٣٦]	$Z_{٣١}Z_{٣٢}Z_{٣٣}Z_{٣٤}Z_{٣٥}Z_{٣٦}$	التكرار ٦
$Z_{١٣}^{-١}$ -$Z_{١٥}$ -$Z_{١٤}Z_{١٦}^{-١}$ $Z_{١١}Z_{١٢}$	$U_{٣٧}U_{٣٨}U_{٣٩}U_{٤٠}U_{٤١}U_{٤٢}$	Z[٣٧..١٠٠;١٢٦..١٢٨;١..٢٩]	$Z_{٣٧}Z_{٣٨}Z_{٣٩}Z_{٤٠}Z_{٤١}Z_{٤٢}$	التكرار ٧
$Z_{٧}^{-١}$ - $Z_{٩}$ -$Z_{٨}Z_{١٠}^{-١}$ $Z_{٥}Z_{٦}$	$U_{٤٢}U_{٤٤}U_{٤٥}U_{٤٦}U_{٤٧}U_{٤٨}$	Z[٣٠..١٢٥]	$Z_{٤٣}Z_{٤٤}Z_{٤٥}Z_{٤٦}Z_{٤٧}Z_{٤٨}$	التكرار ٨
$Z_{١}^{-١}$ -$Z_{٢}$ -$Z_{٣}Z_{٤}^{-١}$	$U_{٤٩}U_{٥٠}U_{٥١}U_{٥٢}$	Z[٢٣..٨٦]	$Z_{٤٩}Z_{٥٠}Z_{٥١}Z_{٥٢}$	التحويل النهائي

٢-٣ – خوارزمية توليد المفتاح:

هي خوارزمية دخلها عبارة عن عدد من البايتات المأخوذة اختيارياً من ملـف توليـد المفاتيح(هو ملف بحجم M byte ١ يتم توليده عشوائياً وإرساله إلى جميع عقد الشبكة بشكل مشفر باستخدام خوارزمية RSA). وانطلاقاً من هذا العـدد الاختيـاري مـن البايتات تقوم هذه الخوارزمية بتوليد مفتاح بطول bit ١٢٨ يستخدم كمفتاح التشـفير في خوارزمية IDEA

وكمثال على هذه الخوارزمية نشرح مبدأ العمل الآتي:

١- بعد أخذ عدد n byte من ملف توليد المفاتيح $a_1,a_2,a_3,\ldots a_n$ نقوم بزيادة عـدد من الأصفار إلى نهاية هذه السلسلة بحيث يصبح العدد قابلاً للقسمة على ١٦ ويتم ذلك كما يأتي:

- تقوم الخوارزمية بحساب القيمة $I = ١٦ - (n \mod ١٦)$

- نضيف I صفراً إلى نهاية السلسلة عندها يصبح عدد البتات قابلاً القسمة على ١٦ أي أن: $(n + I) \mod ١٦ = ٠$

٢- نقسم هذه السلسة من البايتات إلى K سلسلة جزئية حيث $K= (n+I) \text{ div} ١٦$ فتكون كل سلسلة جزئية مكونة من ١٦ byte.

٣- نولد من هذه السلاسل ١٦ مفتاحاً جزئياً بطول ١ byte كما يأتي:

- نأخذ أول بايت من أول سلسلة جزئية ونجري عليها عملية XOR مع أول بايت من كل سلسلة جزئية فنحصل على المفتاح الجزئي الأول، والأمر نفسه يطبق على باقي البايتات فنحصل على ١٦ مفتاحاً جزئياً كل منها مؤلف من ١byte مجموع هذه المفاتيح يشكل المفتاح الكلي المؤلف من ١٢٨ bit.

هذه الخوارزمية بالذات يمكن أن نضيف إليها بعض الإضافات سواء في التصميم أو التنفيذ مما يجعلها خوارزمية خصوصية لأن سرية خوارزمية التشفير تكمن في مفتاح التشفير.

٣- توليد المفتاح Key Generator

يتم توليد المفتاح المؤلف من ١٢٨ bit عن طريق كل من:

- خوارزمية توليد المفتاح - ملف توليد المفاتيح

ويتم ذلك على النحو الآتي:

من ملف توليد المفاتيح نأخذ مجموعة من البتات اختيارياً وذلك حسب قيمة انزياح وطول محددين يتم إرسالهما بشكل مشفر ضمن ترويسة كل رسالة حيث يدل الانزياح على رقم البايت الأولى التي سيتم أخذها من ملف توليد المفاتيح أما الطول فيدل على عدد البايتات المنتقاة

مثال : إذا كان الانزياح = ١٥٠ & الطول = ٢٢٥

هذا يعني أن خوارزمية توليد المفتاح تأخذ بوصفها دخلاً لها البايتات ابتداء من البايت رقم ١٥٠ وحتى البايت رقم ٣٧٤. و هكذا نستطيع اختيار احتمالات هائلة من البايتات كدخل لخوارزمية توليد المفتاح ومن ثَم الحصول على العدد الهائل نفسه من المفاتيح المختلفة المستخدمة في خوارزمية التشفير IDEA.

- لابد من الإشارة إلى أنه يجب وجود مؤشر مع الرسالة يدل على تكامل معلومات الرسالة وأن محتواها لم يتغير خلال عملية النقل سواء كان التغير بشكل مقصود أو غير مقصود.ويمكن تسمية هذا المؤشر بصمة الرسالة، يتم حساب هذه البصمة في طرف الإرسال وإرسالها مع الرسالة المشفرة ويتم حسابها في طرف الاستقبال للتأكد من وصول الرسالة بشكل صحيح. وبصمة الرسالة تحسب وفق عملية رياضية وحيدة الاتجاه تقوم بتحويل كامل معلومات الرسالة المشفرة إلى إطار قصير من المعلومات نطلق عليه كما ذكرنا بصمة الرسالة .إن تغيير بت واحدة في الرسالة يؤدي إلى الحصول على بصمة مختلفة لها. وخوارزمية حساب بصمة الرسالة تكون من نوع التابع وحيد الاتجاه بحيث إذا تمت معرفة بصمة ما فمن غير الممكن استرجاع الرسالة المولدة لتلك البصمة .وغالباً لا يقل طول البصمة عن ١٦ byte.

٤ - مستوى سرية خوارزمية التشفير

تعد كل خوارزميات التشفير المولدة رياضياً قابلة للكسر نظرياً. لكن ما يميزها عن بعضها هو زمن التغطية (الزمن اللازم لعملية الكسر) هذا الزمن الذي يتبع إلى الطاقة الحسابية المتوافرة لدى محلل الشيفرة وإلى أهمية المعلومة مع الزمن. ولكن عند محاولة كسر التشفير يجب مراعاة ما يأتي:

- ألا تتجاوز تكلفة كسر التشفير قيمة المعلومات المشفرة.

- ألا يتجاوز زمن فك التشفير الزمن المفيد للمعلومات أو زمن حياة المعلومات.

وبناء عليه كلما تطورت الطاقة الحسابية للحواسب أدى ذلك إلى تخفيض زمن التغطية الذي يتغير أيضاً حسب أسلوب التحليل المستخدم الذي يمكن أن يكون:

- الأسلوب العشوائي – المعالجة المتوازية

بفرض أن المفتاح نفسه سوف يستخدم لتشفير كل النصوص. فيمكن للمهاجم أن يستخدم الأسلوب العشوائي لكسر التشفير، أي التجربة و الخطأ وفي هذه الحالة يكون عدد الاحتمالات التي يجب تجربتها هي ($٢^{١٢٨}$) محاولة حيث ١٢٨ هو طول المفتاح المستخدم في خوارزميتنا وهذا يساوي إلى $٣,٤ * ١٠^{٣٨}$ محاولة تقريباً

فإذا استطاع المتلصص تجربة مفتاح كل ١ ميكرو ثانية فالوقت المطلوب لتجربة جميع المفاتيح سوف يكون زهاء $١٠^{٢٥}$ سنة.

أما في حال استخدام المعالجة المتوازية باستخدام عدة معالجات تقوم كلها بمعالجة المشكلة آنياً، فإن الرقاقة الواحدة التي تستطيع معالجة مفتاح واحد كل نانو ثانية (إذا كان ذلك ممكناً) تستطيع تجربة زهاء($١٠^{١٤}$) مفتاحاً في اليوم الواحد، أي أنها تحتاج تقريباً إلى($١٠^{٢٤}$) يوماً لتجربة جميع المفاتيح. وهذا يعني أننا نحتاج إلى ($١٠^{٢٤}$) رقاقة تعمل على التوازي بهذا المعدل حتى يتم إنجاز الكشف خلال يوم واحد وبالطبع إنجاز مثل هذه الآلة ليس بالأمر السهل إن لم يكن مستحيلاً.

إن التعقيدات التي ذُكرت كانت من أجل مفتاح واحد يستخدم لتشفير العديد من الرسائل، أما في الخوارزمية المدروسة هنا فإن المفتاح يتغير بشكل دائم وعشوائي وذلك حسب القيمة المختارة من ملف توليد المفاتيح حيث يمكن أن يتغير المفتاح في كل رسالة أي أن لكل رسالة مفتاحها الخاص، ومن ثَم حتى ولو تمكن المتلصص من كشف مفتاح تشفير لأحدى الرسائل فهذا لا يعني أنه تم كسر خوارزمية التشفير. مع العلم أنه حتى يتمكن شخص ما من كشف محتوى الرسالة يجب عليه:

أولاً الحصول على ملف توليد المفاتيح وفك تشفيره. ثانياً يجب عليه معرفة قيمتي الطول و الانزياح المشفرتين بخوارزمية RSA والموضوعتين في ترويسة الرسالة. وكذلك يجب كسر خوارزمية توليد المفتاح وأخيراً كسر خوارزمية التشفيرIDEAفإذا كان من الصعب جداً النجاح بواحدة من تلك الخطوات فكيف يمكن النجاح بكل الخطوات وكسر كل الخوارزميات معاً؟

٥- مراحل تنفيذ برنامج التشفير

قبل البدء بتشفير الرسائل بواسطة الخوارزمية المنفذة نقوم بتوزيع ملف توليد المفاتيح و الذي تم توليده عشوائياً كما ذكرنا نقوم بتوزيعه على جميع فروع الشـبكة مشَـفراً باستخدام خوارزمية RSA وذلك تفادياً لكشف هذا الملف. يوزع هذا الملف مرة واحدة إلا إذا رغبنا في تبديله من فترة إلى أخرى،عندها نقوم بتوليد ملـف عشـوائي آخـر وإرساله بالطريقة نفسها. ويمكننا تغيير حجم هذا الملف لأن حجمه لا يؤثر في توليد المفاتيح. أما التشفير فيتم على الشكل الآتي:

١- نحدد قيمة الانزياح والطول اللازمين لتوليد مفتاح التشفير.

٢- تقوم خوارزمية توليد المفتاح بالاعتماد على ملف توليد المفاتيح وقيمتي الانزياح والطول بتوليد المفتاح العشوائي١٢٨bit الذي سيستخدم في خوارزمية IDEA.

٣- يتم تشفير الرسالة الواضحة بواسطة خوارزمية التشفير IDEA التـي تسـتخدم مفتاح التشفيرالذي تم توليده سابقاً.

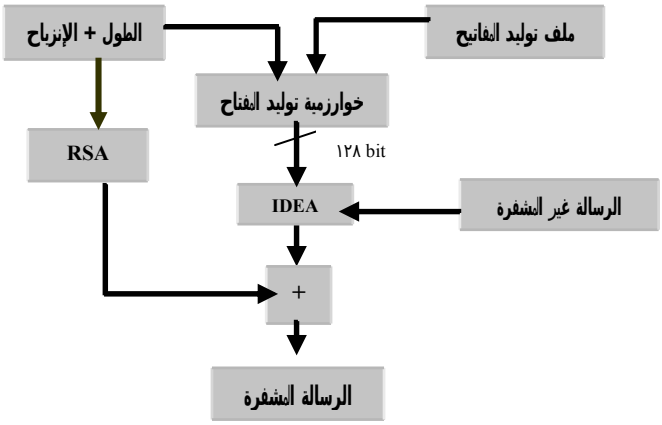
٤- تُشفّر قيمتا الانزياح والطول بواسطة خوارزمية RSA.

٥- يتم إضافة القيمة المشفرة لكلٍ من الانزياح والطـول إلـى ترويسـة الرسـالة المشفرة.

٦- حساب بصمة الرسالة من خلال خوارزمية خاصة بذلك ووضعها في ترويسـة الرسالة.

٧- في النهاية يتم إرسال الرسالة إلى الوجهة المقصودة.

والشكل (٥) يبين البنية الكاملة لهذه الخوارزمية:



شكل (٥)

٥- فك التشفير The Decryption

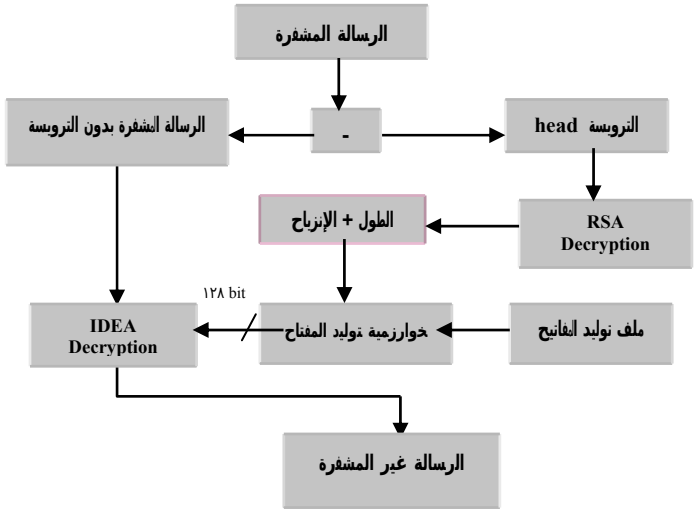
١- بعد أن يتم استقبال الرسالة تقوم خوارزمية فك التشفير بفصل الترويسة عن الرسالة والتأكد من صحتها من خلال بصمة الرسالة.

٢- يتم فك تشفيرجزء الترويسة الذي يحتوي على الطول والانزياح وذلك بواسطة خوارزمية RSA .

٣- تقوم خوارزمية توليد المفاتيح وبالاعتماد على قيمتي الطول والانزياح بتوليد المفتاح ١٢٨ bit.

٤- في النهاية يتم إدخال الرسالة المشفرة إلى خوارزمية IDEA التي تستخدم المفتـاح المولّد ١٢٨ bit للحصول على الرسالة غير المشفرة.

و الشكل (٦) يبين المخطط لمراحل خوارزمية فك التشفير.



شكل(٦)

٦- النتائج:

١- تم اعتماد خوارزمية تتألف من أكثر من خوارزمية جزئية ممـا أدى إلـى رفـع مستوى السرية والتعقيد والتغلب على مشكلة توزيع المفاتيح.

٢- تضمنت الخوارزمية بعض الأفكار الأصلية سواء في التصميم أو التنفيذ مما جعل لها خصوصية.

٣- نظراً لأن السرية تكمن في مفتاح التشفير، فهناك إمكانية توليد المفتـاح و تغيـره بعيداً عن مصمم النظام وذلك لضمان السرية وحصر المسؤولية.

٤- مفاتيح التشفير المستخدمة عشوائية وصعبة الكشف.

References

١- "Network and Internet work security principles and practice"
William Stalling, Ph.D. , prentice-Hall .Inc , New Jersey ,١٩٩٥ .

٢ –"Fundamentals of computer security technology"
Edward G.amoroso, prentice-Hall international .Inc ,
New Jersey, ١٩٩٤.

٣- "Applied cryptography –Protocols" (Algorithms, and Source code in C) Bruce Schneier , John Wily and Sons, Inc, USA ١٩٩٤.

٤ - "الحاسب وأمن المعلومات"، حسن ظاهر داود، الريـاض، مكتبـة الملـك فهـد الوطنية، ٢٠٠٠ م.

تاريخ ورود البحث إلى مجلة جامعة دمشق ٢٠٠٣/٩/٢٠.