الفصل السابع

شفرات تصويب الأخطاء الاندفاعية
Burst Error-Correcting Codes

(٧,١) مقدمة
Introduction

لغاية الآن كان اهتمامنا منصباً على تصميم شفرات تصويب أخطاء موزّعة عشوائياً. ولكن هناك بعض القنوات التي تسمح بحدوث أخطاء قريبة من بعضها بعضاً. على سبيل المثال، من الممكن أن يكون مصدر التشويش على قرص مدمج هو خدش على ذلك القرص ومن ثم فجميع الإحداثيات الواقعة على ذلك الخدش قد تبدلت أو قد تم مسحها مما يتسبب بمجموعة من الأخطاء القريبة بعضها من بعض. كما أن بقع ضوء الشمس تتسبب في وقوع أخطاء قريبة من بعضها بعضاً في الرسائل المرسلة من الأقمار الصناعية إلى الأرض. نُسمي مثل هذه الأخطاء التي تحدث بهذه الصورة أخطاء اندفاعية (أو أخطاء مفاجئة).

لنفرض أنه يمكن تحليل كثيرة الحدود $e(x)$ المقابلة للكلمة e على النحو $e(x) = x^k e'(x)$ حيث $e'(0) = 1$. عندئذ، نقول إن **طول اندفاع e (Burst Length of e)** هو $deg(e'(x)) + 1$. وبهذا فطول الاندفاع هو عدد الإحداثيات من أول وقوع للإحداثي 1 في e إلى آخر وقوع للإحداثي 1 في e.

هناك مفهوم مرادف وهو **طول الاندفاع الدوري (Cyclic Burst Length)** لكلمة e حيث يكون طول الاندفاع الدوري للكلمة $e\epsilon K^n$ يساوي ℓ إذا كانت الدرجة الصغرى لكثيرات الحدود $x^k e(x)(mod 1+x^n)$ حيث $k=0,1,\cdots,n-1$ هي $\ell-1$.

مثال (٧,١,١)

لنفرض أن $n=7$ وأن $e=0101100$. عندئذ،

$$e(x)=x+x^3+x^4=x(1+x^2+x^3)$$

وبهذا تكون $e'(x)=1+x^2+x^3$ و $e'(0)=1$. إذن، طول اندفاع e يساوي $3+1=4$. (لاحظ أنه من الممكن الحصول على طول الاندفاع بعدّ الإحداثيات بين أول وقوع للإحداثي 1 وآخر وقوع للإحداثي 1 في الكلمة e). لايجاد طول الاندفاع الدوري يتوجب علينا حساب $x^k e(x)(mod 1+x^7)$ لكل $k=0,1,2,\cdots,6$ وإيجاد كثيرة الحدود الأصغر درجة من بينها. من السهل أن نرى أن $x^6 e(x)(mod 1+x^7)$ هي كثيرة الحدود الأصغر درجة ودرجتها تساوي 3. إذن، طول الاندفاع الدوري ℓ للكلمة e يحقق $\ell-1=3$. وبهذا يكون $\ell=4$.

أما إذا كانت $e=1000100 \leftrightarrow 1+x^4=x^0(1+x^4)$ فنرى أن طول الاندفاع للكلمة e هو $4+1=5$ ولكن $x^3(1+x^4)\equiv 1+x^3(mod 1+x^7)$ هي كثيرة الحدود الأصغر درجة ومن ثم فطول الاندفاع الدوري للكلمة e هو $\ell=3+1=4$. ▲

لحد الآن افترضنا أن نمط الخطأ الأرجح وقوعه هو نمط الخطأ ذو الوزن الأصغر حيث بنينا هذا الافتراض على أساس أن الأخطاء مُستقلة بعضها عن بعض. ولكن الوضع مختلف في معظم التطبيقات الحقيقية، ولهذا يتحتّم علينا تغيير إستراتيجية تصويب الأخطاء.

عند استخدامنا طريقة MLD للشفرات الخطيّة، اخترنا ممثّل المجموعة المشاركة ليكون الكلمة ذات الوزن الأصغر في تلك المجموعة المشاركة واعتبرنا أن هذه الشفرة

تصوّب الأخطاء من النوع t عندما تقع جميع الكلمات التي وزنها لا يزيد عن t في مجموعات مشاركة مختلفة لتلك الشفرة. ولكن لمعالجة تصويب الأخطاء الاندفاعية، نختار ممثّل المجموعة المشاركة لنمط الخطأ ليكون الكلمة ذات الطول الاندفاعي الأصغر بين كلمات تلك المجموعة المشاركة. ولهذا نقول إن الشفرة الخطيّة **تصوّب الأخطاء الاندفاعية من النوع ℓ (ℓ-Burst Error Correcting Code)** إذا وقعت جميع الكلمات التي طولها الاندفاعي لا يزيد عن ℓ في مجموعات مشاركة مختلفة لتلك الشفرة. بصورة عامة، إذا كانت C تصوّب أخطاء من النوع t وتصوّب أخطاء اندفاعية من النوع ℓ فإن $t \leq \ell$ (لماذا ؟) ومن الممكن أن تكون هذه المتباينة فعلية (انظر التمارين **(٧,١,٥)**، **(٧,١,٦)**، **(٧,١,٧)**). بصورة مماثلة نقول إن الشفرة الخطيّة **تصوّب أخطاء اندفاعية دورية من النوع ℓ (ℓ-Cyclic Burst Error Correcting Code)** إذا وقعت جميع الكلمات التي طولها الاندفاعي الدوري لا يزيد عن ℓ في مجموعات مشاركة مختلفة لتلك الشفرة.

مثال (٧,١,٢)

اعتبر جميع أنماط الأخطاء الاندفاعية الدورية غير الصفرية التي طولها لا يزيد عن 3 في K^{15}. كل نمط خطأ من هذه الأنماط هو على الصورة $e(x) = x^k e'(x)$ ، $k = 0,1,\cdots,14$ حيث $e'(x) \epsilon \{1,1+x,1+x^2,1+x+x^2\}$. ولهذا يكون عدد أنماط الأخطاء هذه هو $4 \times 15 = 60$. ▲

مثال (٧,١,٣)

افرض أن $g(x) = 1 + x + x^2 + x^3 + x^6$ كثيرة حدود مولّدة لشفرة خطيّة دورية من الطول 15 والبُعد 9. من الواضح أن هذه الشفرة لا تصوّب 3 أخطاء ؛ وذلك لوجود 576 كلمة من وزن لا يزيد عن 3 وعدد المجموعات المشاركة يساوي 64 فقط. ولكن عدد أنماط الخطأ التي طول اندفاعها الدوري لا يزيد عن 3 يساوي 61 (انظر المثال **(٧,١,٢)**). ولذا من المحتمل أن تصوّب هذه الشفرة أخطاء اندفاعية

من النوع 3 (في الحقيقة هي كذلك، انظر التمرين (**٧,١,٤**)). حيث يمكن التحقق من ذلك بحساب تناذرات $x^k e'(x)(mod\ g(x))$ ، $k = 0,1,\cdots,14$ حيث $e'(x)\epsilon\{1,1+x,1+x^2,1+x+x^2\}$. ▲

تمارين

(**٧,١,٤**) تحقق من أن أنماط الأخطاء الاندفاعية الدورية ذات الطول 3 في K^{15} تنتمي إلى مجموعات مشاركة مختلفة للشفرة المقدمة في المثال (**٧,١,٣**).

(**٧,١,٥**) أثبت أن $g(x) = 1 + x^2 + x^4 + x^5$ تولّد شفرة خطيّة دورية C من الطول 15 وتصوّب أخطاء اندفاعية دورية من النوع 2. هل تصوّب C أخطاء من النوع 2 ؟

(**٧,١,٦**) أثبت أن $g(x) = 1 + x^3 + x^4 + x^5 + x^6$ تولّد شفرة خطيّة دورية C من الطول 15 وتصوّب أخطاء اندفاعية دورية من النوع 3. هل تصوّب C أخطاء من النوع 3 ؟ (**إرشاد**: استخدم حد هامينغ).

(**٧,١,٧**) أثبت أن $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ تولّد شفرة خطيّة دورية من الطول 15 وتصوّب أنماط أخطاء من النوع 2 وتصوّب أيضاً أنماط أخطاء اندفاعية دورية من النوع 4.

إذا كانت الشفرة C تصوّب أخطاء من النوع t وتصوّب أخطاء اندفاعية من النوع ℓ فلقد لاحظنا سابقاً أن $\ell \geq t$. تقدم لنا المبرهنة التالية حداً أعلى لقيمة ℓ. من الممكن تقديم حد أعلى أفضل من الحد الأعلى الذي تقدمه المبرهنة (انظر التمرين (**٧,١,١٠**)) ولكن الحد الأعلى الذي تقدمه هذه المبرهنة يفي بالغرض.

مبرهنة (٧,١,٨)

إذا كانت C شفرة خطيّة من الطول n والبُعد k وتصوّب أخطاء اندفاعية من النوع ℓ فإن $\ell \leq n - k$.

البرهان

لنفرض أن C شفرة خطيّة من النوع (n,k) وتصوّب أنماط أخطاء اندفاعية من النوع ℓ. عندئذ، تقع جميع أنماط الأخطاء الاندفاعية ذات الطول الذي لا يزيد عن ℓ في مجموعات مشاركة مختلفة. من ذلك نرى عدم وجود كلمتين حيث أول ℓ من إحداثيات كل منها يساوي 1 بحيث تقعان في مجموعة مشاركة واحدة. ولكن عدد هذه الكلمات يساوي 2^{ℓ}. ومن ثم عدد المجموعات المشاركة هو على الأقل 2^{ℓ}. وبهذا يكون $\ell \leq n-k$. ■

تمارين

(٧,١,٩) تحقق من أن الشفرات المقدمة في التمارين **(٧,١,٥)**، **(٧,١,٦)**، **(٧,١,٧)** تحقق الحد الأعلى المقدم في المبرهنة **(٧,١,٨)**.

(٧,١,١٠) إذا كانت C شفرة خطيّة من النوع (n,k) وتصوّب أنماط أخطاء اندفاعية من النوع ℓ فأثبت أن $\ell \leq (n-k)/2$ [**إرشاد:** أثبت إمكانية كتابة أي نمط خطأ اندفاعي من الطول 2ℓ كمجموع نمطي خطأ بطول اندفاعي $e_1 \leq \ell$ و $e_2 \leq \ell$ على التوالي ومن ثم أثبت أن $e_1 + e_2 \notin C$].

إذا كانت C شفرة خطيّة دورية فتوجد خوارزمية فك تشفير فعّالة لتصويب أنماط الأخطاء الاندفاعية الدورية. لنفرض إذن أن C تصوّب أنماط أخطاء اندفاعية دورية من النوع ℓ ومولّدة بكثيرة حدود $g(x)$ من الدرجة $n-k$. من النقاش المقدم قبل المثال **(٧,٣,٤)** نرى أن:

$$H = \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{n-1} \end{bmatrix} = \begin{bmatrix} I_{n-k} \\ r_{n-k} \\ r_{n-k+1} \\ \vdots \\ r_{n-1} \end{bmatrix}$$

هي مصفوفة اختبار النوعية للشفرة C حيث $1 \leq i \leq n-1$ ، r_i هي الكلمة من الطول $n-k$ التي تقابل كثيرة الحدود $r_i(x) \equiv x^i (mod\ g(x))$. ولقد بيّنا أيضاً

(باستخدام المصفوفة H) أنه إذا كانت $w(x) = c(x) + e(x)$ هي الكلمة المستقبلة فيكون التناذر للكلمة $w \leftrightarrow w(x)$ هو:

$$wH = s \leftrightarrow s(x) \equiv w(x)(mod\ g(x))$$

الحقيقتان التاليتان هما السبب الذي يجعل استخدام C و H فعّالاً عند فك تشفير أنماط الأخطاء الاندفاعية الدورية:

(١) إذا كان $e \leftrightarrow e(x)$ نمط خطأ طوله الاندفاعي لا يزيد عن ℓ فنرى استناداً إلى المبرهنة (**٧,١,٨**) أن $\ell \le n - k$. ولذا يوجد i ، في الفترة $0 \le i \le n-1$ بحيث تقع جميع الإحداثيات 1 من الإزاحة الدورية e_i للكلمة e في البداية (أي أن $x^i e(x)(mod 1 + x^n) \leftrightarrow e_i$ ذات درجة أصغر من ℓ).

(٢) من السهل حساب تناذر الإزاحة الدورية w_i للكلمة w لأن:

$$\begin{aligned} s_i &= w_i H \\ &\leftrightarrow (x^i w(x)(mod 1 + x^n)) mod g(x) \\ &\equiv x^i w(x)(mod g(x)) \qquad (\text{لأن } g(x) \text{ تقسم } 1 + x^n) \\ &\equiv x^i s(x)(mod\ g(x)) \end{aligned}$$

ولهذا يكون تصويب الخطأ $w = c + e$ على النحو التالي:

لكل i ، $0 \le i \le n-1$ نقوم بحساب التناذرات $s_i \leftrightarrow x^i s(x)(mod\ g(x))$ للكلمة w_i بالتالي حتى نجد واحداً وليكن s_j يحقق $deg(s_j(x)) < l$ حيث استخدمنا الحقيقة (٢) في هذه الحسابات. الآن، بما أن أول $n-k$ من صفوف H هي المصفوفة المحايدة فنرى أن تناذر الكلمة $e_j = s_j 00\cdots 0$ (التي نحصل عليها من s_j باضافة k من الأصفار إلى s_j) هو $e_j H = s_j$. وبما أن C دورية فنعلم أن $w_j = c_j + e_j$ حيث c_j هي كلمة الشفرة التي نحصل عليها من الإزاحة j الدورية للكلمة c. ومن ثم نستطيع إزاحة e_j إلى الخلف بعدد j من الإزاحات الدورية لنحصل على $e = e_0$. وهذا هو نمط الخطأ الاندفاعي الدوري. من ذلك نحصل على الخوارزمية التالية:

خوارزمية (٧,١,١١) [فك تشفير أنماط أخطاء اندفاعية دورية]

لنفرض أن w كلمة مُستقبلة تم تشفيرها باستخدام شفرة خطيّة دورية تصوّب أنماط أخطاء اندفاعية دورية من النوع ℓ حيث $g(x)$ هي كثيرة حدود مولّدة للشفرة.

(١) احسب كثيرة حدود التناذر $s(x) \equiv w(x)(mod\ g(x))$.

(٢) لكل $i \geq 0$ ، احسب $s_i(x) \equiv x^i s(x)(mod\ g(x))$ حتى تجد كثيرة حدود تناذر $s_j(x)$ تحقق $deg(s_i(x)) \leq \ell - 1$.

عندئذ، يكون نمط الخطأ الاندفاعي الدوري الأرجح وقوعاً هو:

$$e(x) \equiv x^{n-j} s_j(x)(mod 1 + x^n).$$

مثال (٧,١,١٢)

لنفرض أن $g(x) = 1 + x + x^2 + x^3 + x^6$ كثيرة حدود مولّدة لشفرة خطيّة دورية من الطول 15 وتصوّب أنماط أخطاء اندفاعية دورية من النوع 3. استخدم الخوارزمية **(٧,١,١١)** لفك تشفير الكلمة المستقبلة 111100100001010 على افتراض أرجحية وقوع أنماط أخطاء اندفاعية دورية.

الحل

(١)
$$s(x) \equiv 1 + x + x^2 + x^3 + x^6 + x^{11} + x^{13}(mod\ g(x))$$
$$\equiv 1 + x^3 + x^4 + x^5$$

(٢)
$$s_1(x) \equiv xs(x)\big(mod\ g(x)\big) = 1 + x^2 + x^3 + x^4 + x^5$$
$$s_2(x) \equiv x^2 s(x)\big(mod\ g(x)\big) = 1 + x^2 + x^4 + x^5$$
$$s_3(x) \equiv x^3 s(x)\big(mod\ g(x)\big) = 1 + x^2 + x^5$$
$$s_4(x) \equiv x^4 s(x)\big(mod\ g(x)\big) = 1 + x^2$$

وإن $deg(s_4(x)) = 2 \leq \ell - 1$. إذن، نمط الخطأ الأرجحي هو:

$$e(x) \equiv x^{15-4} s_4(x)(mod(1 + x^{15}))$$
$$= x^{11} + x^{13}$$

ومن ثم تكون كلمة الشفرة الأرجح هي :

$$c(x) = w(x) + e(x) = 1 + x + x^2 + x^3 + x^6$$

$$\leftrightarrow 111100100000000. \quad \blacktriangle$$

تمارين

(٧,١,١٣) لنفرض أن $g(x) = 1 + x + x^2 + x^3 + x^6$ كثيرة حدود مولّدة لشفرة خطيّة دورية C من الطول 15 وتصوّب أنماط أخطاء اندفاعية دورية من النوع 3. فُك تشفير كل من الكلمات المستقبلة التالية التي شُفّرت بواسطة الشفرة C :

(أ) 101101110001000 (ب) 001101100010101

(ج) 100110101010011 (د) 101101000010111

(هـ) 000000111110000.

(٧,١,١٤) افرض أن $g(x) = 1 + x^2 + x^4 + x^5$ كثيرة حدود مولّدة لشفرة خطيّة دورية C من الطول 15 وتصوّب أنماط أخطاء اندفاعية دورية من النوع 2. فُك تشفير كل من الكلمات المستقبلة التالية التي شُفّرت بواسطة الشفرة C :

(أ) 010101000010010 (ب) 011010010010100

(ج) 001101000000100 (د) 000100010100101

(هـ) 000000011111001.

تتمتع شفرات ريد وسولومن بقدرة جيدة على تصويب الأخطاء الاندفاعية. تذكر أنه إذا كانت C هي الشفرة $RS(2^r, \delta)$ فإن $\hat{C}$ هي التمثيل الثنائي للشفرة C (انظر المثال **(٦,٢,٦)**).

مبرهنة (٧,١,١٥)

لتكن C هي الشفرة $RS(2^r, 2t + 1)$. عندئذ، $\hat{C}$ شفرة تصوّب أنماط أخطاء اندفاعية من النوع ℓ حيث $\ell \geq r(t - 1) + 1$.

البرهان

ينتج عن أي نمط خطأ اندفاعي e طوله لا يزيد عن $r(t-1)+1$ كلمة $\widehat{w}=\hat{c}+e$ حيث $d(w,c)\leq t$. إذن، يكون فك تشفير w هي كلمة الشفرة $c\epsilon RS(2^r,2t+1)$. وبهذا تكون $\hat{c}\epsilon\hat{C}$ هي أقرب كلمة شفرة للكلمة $\widehat{w}$. ■

تستخدم شفرات ريد وسولومن في الأقراص الممغنطة حيث تتسبب الخدوش على القرص بحدوث أخطاء اندفاعية. كما أنها تستخدم أيضاً في الاتصالات الفضائية من قبل NASA و ESA حيث تتسبب بقع ضوء الشمس بحدوث أخطاء اندفاعية أثناء عملية الإرسال التي تكون على شكل موجات كهرومغناطيسية. وفي كلتا الحالتين يفضّل أن نفترض أن الأخطاء التي وقعت هي أخطاء اندفاعية وليست أخطاء عشوائية.

مثال (٧,١,١٦)

تصوّب الشفرة $RS(8,5)$ المقدمة في التمرين (٦,٢,٨) جميع الأخطاء الاندفاعية التي طولها لا يزيد عن $r(t-1)+1=4$. ▲

(٧,٢) التوريق البيني

Interleaving[1]

إحدى طرق تحسين قدرة الشفرات على تصويب الأخطاء الاندفاعية هي استخدام تقنية التوريق البيني حيث تكمن فكرة هذه التقنية باعادة ترتيب إرسال إحداثيات كلمة الشفرة. الطريقة التي اتبعناها حتى الآن في إرسال الرسائل $m_1,m_2,\cdots$ كانت عبارة عن تشفير هذه الرسائل إلى كلمات شفرة مقابلة $c_1,c_2,\cdots$ ومن ثم إرسال كلمات الشفرة واحدة بعد الأخرى بهذا الترتيب. لنفرض الآن أننا قمنا باختيار أول s كلمة من

(١) المترجمان: الترجمة الحرفية للكلمة interleave هي يورق بينياً. أي يضع ورقة بيضاء بين ورقتي كتاب. ولهذا نرى أنها ترجمة مناسبة لموضوع هذا البند.

كلمات الشفرة ثم بعد ذلك قمنا بإرسال أول إحداثي من كل كلمة من هذه الكلمات، بعد ذلك قمنا بإرسال ثاني إحداثي من كل كلمة من هذه الكلمات وهكذا. وبمجرد الانتهاء من إرسال الإحداثيات التي عددها ns من أول s كلمة من كلمات الشفرة بالترتيب المبيّن نقوم باختيار مجموعة جديدة من كلمات الشفرة عددها s ونكرّر عملية إرسال الإحداثيات بالترتيب نفسه للمجموعة الأولى. وهكذا إلى أن ننتهي من عملية الإرسال. تُسمى إعادة ترتيب إحداثيات كلمات الشفرة بهذا الأسلوب، **التوريق البيني بعمق s (Interleaving to Depth s)**. يمكن صياغة التوريق البيني لكلمات الشفرة $c_1, c_2, \cdots$ لعمق s على النحو التالي:

لكل $i = 0,1,2, \cdots$ نقوم بإرسال إحداثيات كلمة الشفرة بالترتيب التالي:

$$c_{is+1}, 1, c_{is+2}, 1, \cdots, c_{is+s}, 1, c_{is+1}, 2, c_{is+2}, 2, \cdots, c_{is+s}, 2, \cdots, c_{is+1}, n, \cdots, c_{is+s}, n,$$

ولتسهيل رؤية عملية الإرسال هذه نقوم بكتابة كلمات الشفرة $c_{is+1}, \cdots, c_{is+s}$ على شكل صفوف (انظر الجدول (٧,١)) ثم ارسال الإحداثيات عموداً عموداً.

الجدول (٧,١). توريق بيني لعمق s.

$c_{is+1,1}$	$c_{is+1,2}$	$c_{is+1,3}$	$\cdots$	$c_{is+1,n}$
$c_{is+2,1}$	$c_{is+2,2}$	$c_{is+2,3}$	$\cdots$	$c_{is+2,n}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$c_{is+s,1}$	$c_{is+s,2}$	$c_{is+s,3}$	$\cdots$	$c_{is+s,n}$

مثال (٧,٢,١)

لتكن C شفرة خطية ذات مصفوفة مولّدة $G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$. إذا لم يستخدم التوريق البيني فيتم ارسال كلمات الشفرة التالية:

$$c_1 = 100110, \quad c_4 = 010101$$
$$c_2 = 010101, \quad c_5 = 100110$$
$$c_3 = 111000, \quad c_6 = 111000$$

كلمة بعد الأخرى ومن ثم فإحداثيات الشفرة تُرسل بالترتيب التالي :

.100110 010101 111000 010101 100110 111000

أما إذا استخدمنا التوريق البيني لعمق 3 فنقوم بإرسال الإحداثيات الأولى من كلمات الشفرة c_1, c_2, c_3 أولاً (أي 101) وبعد ذلك نُرسل الإحداثي الثانية من كلمات الشفرة c_1, c_2, c_3 (أي 011) وهكذا. وبهذا يتم إرسال إحداثيات الكلمات c_1, c_2, c_3 على النحو التالي :

.011 101 001 110 010 100 ▲

ما هو تأثير التوريق البيني لعمق s على قدرة تصويب الشفرة C لأنماط الأخطاء الاندفاعية؟ لرؤية ذلك، نفرض أن ترتيب إرسال الإحداثي الأول من كلمة الشفرة c هو i. حينئذ، تكون مواقع بقية إحداثيات الكلمة c هي $i + s, i + 2s, \cdots, i + (n-1)s$. لنفرض أن C شفرة تصوّب أخطاء اندفاعية من النوع ℓ. إذا استخدمنا التوريق البيني لعمق s للشفرة C فنرى إن أي اندفاع للأخطاء أثناء عملية الإرسال طوله لا يزيد عن $s\ell$ ينتج عنه نمط خطأ اندفاعي في كلمة الشفرة c طوله لا يزيد عن ℓ، وبهذا يكون فك تشفير c صحيحاً بحالة عدم وجود أنماط أخطاء اندفاعية أخرى تؤثر في c. وبهذا نكون قد برهنا النتيجة التالية :

مبرهنة (٧,٢,٢)

لتكن C شفرة تصوّب أنماط أخطاء اندفاعية من النوع ℓ. إذا تم توريق C بينياً لعمق s فإنه يتم تصويب جميع أنماط الأخطاء الاندفاعية التي طولها لا يزيد عن $s\ell$ بافتراض أن كل كلمة شفرة تأثّرت على الأكثر باندفاع أخطاء واحد. ■

ملحوظة

إن الشرط الاحترازي بعدم تأثر كل من كلمات الشفرة بأكثر من اندفاع واحد من الأخطاء يفترض وجود مسافة كافية بين كل نمطين من أنماط الأخطاء الاندفاعية

أثناء عملية الإرسال لتجنب تأثير نمطين من الأخطاء الاندفاعية على قالب واحد طوله s من كلمات الشفرة. ولهذا فإن اختيار عدد كبير s يزيد من ضمان تصويب أنماط الأخطاء الاندفاعية تحت شروط المبرهنة (**٧,٢,٢**)، كما أنه يضمن وجود مسافة كافية بين أنماط الأخطاء الاندفاعية أثناء الإرسال.

مثال (٧,٢,٣)

تصوّب الشفرة C المقدمة في المثال (**٧,٢,١**) خطأ واحداً. إذا تم توريقها بينياً لعمق 3 فهذا يزيد من قدرتها بحيث تستطيع تصويب أنماط أخطاء اندفاعية من الطول 3. ▲

تمارين

(**٧,٢,٤**) شفّر الرسائل $m_1 = 1000$ ، $m_2 = 0110$ ، $m_3 = 1110$ ، $m_4 = 0011$ ، $m_5 = 0110$ ، $m_6 = 0001$. جد أيضاً الإحداثيات المرسلة إذا تم توريق الشفرة لعمق s حيث:

$$G = \begin{bmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{bmatrix}$$

(أ) $s = 1$ (ب) $s = 2$ (ج) $s = 3$.

(**٧,٢,٥**) ذكرنا في المثال (**٧,١,٣**) أن كثيرة الحدود $g(x) = 1 + x + x^2 + x^3 + x^6$ تولّد شفرة خطية دورية C من الطول 15 ولها القدرة على تصويب أنماط أخطاء اندفاعية دورية من النوع 3. استخدم المصفوفة G المولّدة للشفرة C:

$$G = \begin{bmatrix} 111100100000000 \\ 011110010000000 \\ \vdots \\ 000000001111001 \end{bmatrix}$$

لتشفير الرسائل $m_1(x) = 1$ ، $m_2(x) = x^2$ ، $m_3(x) = 1 + x$ ، $m_4(x) = 1 + x^2$ ، $m_5(x) = x^3$ ، $m_6(x) = 1$. جد الإحداثيات المرسلة إذا تم توريق C بينياً لعمق s حيث:

(أ) $s = 1$ (ب) $s = 2$ (ج) $s = 3$.

في كل من قيم S المعطاة، استخدم المبرهنة (**٧,٢,٢**) لتجد أي أنماط أخطاء اندفاعية تستطيع الشفرة تصويبها.

أحد عوائق التوريق البيني لعمق s هو ضرورة تشفير عدد s من كلمات الشفرة قبل الشروع في إرسال أي منها. للتغلب على هذا العائق نستخدم **الاطار المؤجل للتوريق البيني من النوع s (*s*-Frame Delayed Interleaving)** ؛ وذلك بسرد إحداثيات كل من كلمات الشفرة كما هو مبيّن في الجدول (**٧,٢**) (قارن ذلك مع الجدول (**٧,١**)). ومن ثم نقوم بارسال الإحداثيات عموداً عموداً. يحتوي صفيف الجدول (**٧,٢**) على عدد n من الصفوف. لكل كلمة شفرة c_i يوجد إحداثي واحد فقط $c_{i,j}$ في الصف j $(1 \le i \le n)$ حيث $c_{i,j+1}$ تقع في الصف الذي يقع مباشرة أسفل الصف الذي يقع فيه الإحداثي $c_{i,j}$ وتبعد عدد s من الأعمدة عن العمود الواقع فيه الإحداثي $c_{i,j}$ $(1 \le j \le n-1)$.

الجدول (٧,٢). اطار مؤجل للتوريق البيني من النوع s.

$c_{1,1}c_{2,1}\cdots$	$c_{s+1,1}$	$c_{s+2,1}\cdots$	$c_{2s+1,1}$	$c_{2s+2,1}$	$\cdots$	$c_{(n-1)s+1,1}$ $\cdots$
	$c_{1,2}$	$c_{2,2}\cdots$	$c_{s+1,2}$	$c_{s+2,2}$	$\cdots$	$c_{(n-2)s+1,2}$ $\cdots$
			$c_{1,3}$	$c_{2,3}$	$\cdots$	$c_{(n-3)s+1,3}$ $\cdots$
						$\vdots$
						$c_{1,n}$ $\cdots$

من الواضح أن استخدام الاطار المؤجل للتوريق البيني من النوع s يحتاج إلى بعض التحضير؛ لأنه إذا كان $s \ge 1$ فإن العمود الأول من الجدول (**٧,٢**) يحتوي فقط إحداثياً واحداً هو $c_{1,1}$، ولضمان وجود عدد n من الإحداثيات في كل من أعمدة الجدول (**٧,٢**) نقوم بوضع الإحداثي 0 في المواضع الخالية من الإحداثيات. المثال التالي يوضح ذلك مع ملاحظة وضع * عوضاً عن 0 في المواضع الخالية؛ وذلك لتفريقها عن الإحداثي 0 من كلمة الشفرة.

مثال (٧,٢,٦)

لتكن $c_1, c_2, \cdots, c_6$ هي كلمات الشفرة المبينة في المثال (**٧,٢,١**). إذا استخدمنا إطاراً مؤجلاً للتوريق البيني من النوع 1 فيكون الجدول (**٧,٢**) على النحو التالي:

$$
\begin{array}{cccccccccccc}
1 & 0 & 1 & 0 & 1 & 1 & \cdots & & & & & \\
* & 0 & 1 & 1 & 1 & 0 & 1 & \cdots & & & & \\
* & * & 0 & 0 & 1 & 0 & 0 & 1 & \cdots & & & \\
* & * & * & 1 & 1 & 0 & 1 & 1 & 0 & \cdots & & \\
* & * & * & * & 1 & 0 & 0 & 0 & 1 & 0 & \cdots & \\
* & * & * & * & * & 0 & 1 & 0 & 1 & 0 & 0 & \cdots
\end{array}
$$

حيث قمنا بداية بوضع * في المواقع الخالية من الأعمدة. ومن ثم يتم الإرسال على النحو التالي:

$$\cdots ** 0101 *** 110 **** 00 ***** 1.$$

أما إذا استخدمنا إطاراً مؤجلاً للتوريق البيني من النوع 2 فيكون الجدول (**٧,٢**) على النحو التالي:

$$
\begin{array}{cccccccccccccc}
1 & 0 & 1 & 0 & 1 & 1 & \cdots & & & & & & & \\
* & * & 0 & 1 & 1 & 1 & 0 & 1 & \cdots & & & & & \\
* & * & * & * & 0 & 0 & 1 & 0 & 0 & 1 & \cdots & & & \\
* & * & * & * & * & * & 1 & 1 & 0 & 1 & 1 & 0 & \cdots & \\
* & * & * & * & * & * & * & * & 1 & 0 & 0 & 0 & 1 & 0 & \cdots \\
* & * & * & * & * & * & * & * & * & * & 0 & 1 & 0 & 1 & 0 & 0 & \cdots
\end{array}
$$

ومن ثم يتم الإرسال على النحو التالي:

$$\cdots *** 110 **** 01 **** 10 ***** 0 ***** 1. \qquad \blacktriangle$$

المبرهنة التالية هي رديف المبرهنة (**٧,٢,٢**) في حالة استخدام اطار مؤجل للتوريق البيني من النوع s.

مبرهنة (٧,٢,٧)

لتكن C شفرة تصوّب أخطاء اندفاعية من النوع ℓ. إذا كانت C تستخدم إطاراً مؤجلاً للتوريق البيني من النوع s فإن C تصوّب جميع الأخطاء الاندفاعية من النوع $\ell(sn+1)$ بشرط أن تكون كل كلمة شفرة قد تأثرت على الأكثر باندفاع واحد من الأخطاء.

تمارين

(**٧,٢,٨**) استخدم إطاراً مؤجلاً للتوريق البيني من النوع s وكلمات الشفرة المقدمة في التمرين (**٧,٢,٤**) لايجاد الإحداثيات المرسلة عندما يكون

(أ) $s=1$ (ب) $s=2$.

(**٧,٢,٩**) إذا استخدم اطار مؤجل للتوريق البيني من النوع 0 فما هي الإحداثيات المرسلة ؟

(**٧,٢,١٠**) أثبت المبرهنة (**٧,٢,٧**).

عند التطبيق العملي تستخدم شفرتان لتشفير الرسائل. على سبيل المثال، تستخدم شفرتان لتشفير النغمات الموسيقية على الأقراص المغنطة (انظر البند (**٧,٣**)) والشفرتان هما شفرات ريد وسولومن. كما تستخدم كل من NASA و ESA شفرتين إحداهما شفرة ريد وسولومن والأخرى شفرة تلاف (انظر البند (**٨,٢**))، وكما سنرى الآن، يلعب التوريق البيني لعمق s أهمية خاصة في مثل عمليات التشفير هذه المكوّنة من خطوتين.

افرض أن C_1 شفرة خطية من النوع (n_1, k_1, d_1) و C_2 شفرة خطية من النوع (n_2, k_2, d_2). يتم استخدام التوريق البيني في تشفير C_1 و C_2 على النحو التالي:

تُشفّر الرسائل أولاً باستخدام C_1 ومن ثم يستخدم التوريق البيني لعمق k_2 على كلمات الشفرات الناتجة. طول كل من الأعمدة الناتجة عن عملية التوريق البيني هذه هو k_2 (كما في الجدول (**٧,١**)) وبهذا ينظر إليها على أنها رسائل يتم تشفيرها باستخدام C_2. نقوم الآن بتوريق بيني لكلمات الشفرة الناتجة عن التشفير الثاني لعمق s أو لإطار مؤجل من النوع s.

الميزة الأساسية للتشفير بخطوتين هي:

يمكن استخدام C_2 لاكتشاف أخطاء عددها $d_2 - 1$ عوضاً عن استخدامها لتصويب أخطاء. عند اكتشاف أخطاء في احدى كلمات الشفرة C_2 نقوم بتعليم جميع إحداثيات هذه الكلمة ونتعامل معها على أنها إحداثيات غير صحيحة. بعد ذلك نركّز اهتمامنا على كلمات الشفرة C_1. إذا علمنا وجود $n_1 - d_1 + 1$ إحداثياً صحيحاً من إحداثيات كلمة شفرة $c \epsilon C_1$ فباستطاعتنا دائماً إيجاد بقية الإحداثيات التي عددها $d_1 - 1$. يرجع السبب وراء ذلك لاستحالة اتفاق كلمة أخرى من كلمات الشفرة C_1 مع الكلمة c بإحداثيات صحيحة عددها $n_1 - d_1 + 1$؛ لأن جميع كلمات الشفرة تختلف على الأقل بمواقع عددها d_1. إذن، إذا احتوت كل من كلمات C_1 على عدد من الإحداثيات المعلّمة لا يزيد عن $d_1 - 1$ وإذا افترضنا أن جميع الإحداثيات الخاطئة قد تم تعليمها فنرى أنه قد تم فك تشفير كلمات الشفرة بصورة صحيحة.

مثال (٧,٢,١١)

لنفرض أن C_1 و C_2 شفرتان مصفوفتاهما المولدتان هما على التوالي:

$$G_1 = \begin{bmatrix} 10001110 \\ 01001101 \\ 00101011 \\ 00010111 \end{bmatrix} \text{ و } G_2 = \begin{bmatrix} 100110 \\ 010101 \\ 001011 \end{bmatrix}.$$

عندئذ، $(n_1, k_1, d_1) = (8,4,4)$ و $(n_2, k_2, d_2) = (6,3,3)$. سنقوم بتشفير الرسائل $m_1 = 1000$ ، $m_2 = 1100$ ، $m_3 = 1010$ باستخدام التوريق البيني بين C_1 و C_2 حيث ورقت C_2 بينياً لعمق $s = 3 = d_1 - 1$. باستخدام C_1 لتشفير m_1 ، m_2 ، m_3 نرى أن:

$$c_1 = m_1 G_1 = 10001110$$
$$c_2 = m_2 G_1 = 1100011$$
$$c_3 = m_3 G_1 = 10100101$$

بتوريق كلمات الشفرة هذه بينياً لعمق $k_2 = 3$ تكون الرسائل الناتجة عن أعمدة هذا التوريق هي:

.111,010,001,000,100,101,110,011

نستخدم الآن C_2 لتشفير هذه الرسائل لينتج عن ذلك 8 كلمات شفرة يتم توريقها بينياً لعمق $s = 3$ لينتج عن ذلك:

$c_1' = 111000$ $c_4' = 000000$ $c_7' = 110011$

$c_2' = 010101$ $c_5' = 100110$ $c_8' = 011110$

$c_3' = 001011$ $c_6' = 101101$

(تورق c_7' و c_8' مع أول كلمة شفرة c_9' التي تنتج عن الرسائل الثلاث التي تلي ذلك m_4, m_5, m_6). إذن، تكون بداية الإحداثيات المرسلة هي:

.100 110 101 010 001 011 011 000 001 011 010 001 ⋯

ولرؤية كيفية فك التشفير، نفرض أنه قد حصل خطأ في الإحداثيات الستة الأولى. أي أننا استقبلنا:

011 001 101 010 001 011 000 ⋯

بإلغاء تأثير التوريق البيني لعمق $s = 3$ ينتج عن ذلك كلمات مستقبلة طولها $n_2 = 6$:

001000,100101,111011

(لاحظ أنه بالمقارنة مع c_1' ، c_2' ، c_3' على التوالي نرى أن كل منها يحتوي على أخطاء في أول موقعين). الآن، تكتشف C_2 الأخطاء في جميع الكلمات الثلاث هذه (أثبت أن التناذر wH_2 لكل من هذه الكلمات المستقبلة w لا يساوي صفراً حيث H_2 هي مصفوفة اختبار النوعية للشفرة C_2). وبهذا تكون الإحداثيات الـ 18 جميعها معلّمة (نستبدل كل منها بالعلامة *). بفرض عدم اكتشاف أخطاء أخرى بعد عملية مماثلة لإحداثيات كلمات الشفرة $c_4', c_5', \cdots, c_8'$ فنرى بعد إزالة تأثير التوريق البيني لعمق $k_3 = 3$ أننا قد حصلنا على ثلاث كلمات طول كل منها $n_1 = 8$:

$$c_1 = *** 01110$$
$$c_2 = *** 00011$$
$$c_3 = *** 00101$$

بعد ذلك توجد طريقة واحدة فقط للتعويض عن الإحداثيات المعلّمة * بأحد الإحداثيين 0 و 1 للحصول على كلمات شفرة c_1, c_2, c_3. لاحظ أن كلاً من الكلمات الثلاث السابقة تحتوي على إحداثيات معلّمة عددها $d_1 - 1 = 3$. ▲

تمارين

(**٧,٢,١٢**) استخدم الشفرتين C_1 و C_2 لتشفير مجموعات الرسائل التالية مُستخدماً التوريق البيني بينهما إذا علمت أن التوريق البيني للشفرة C_2 هو لعمق s.

(أ) $m_1 = 0110, m_2 = 1011, m_3 = 1111, s = 2$

(ب) $m_1 = 0110, m_2 = 1011, m_3 = 1111, s = 3$

(ج) $m_1 = 0010, m_2 = 1111, m_3 = 1010, s = 3$

(د) $m_1 = 1000, m_2 = 0100, m_3 = 0010,$
$m_4 = 0001, m_5 = 0011, m_6 = 0100, s = 3$

(**٧,٢,١٣**) إذا علمت أن الإحداثيات التالية قد تم تشفيرها بتوريق بيني للشفرتين C_1 و C_2 المقدمتين في المثال (**٧,٢,١١**) حيث ورقت C_1 بينياً لعمق 3 فجد فك تشفير هذه الإحداثيات وذلك بإيجاد الرسائل m_1, m_2, m_3.

(أ) $0000010011101100010001110001110001110000000000000000\cdots$

(ب) $1000110011111010100110011110101001101001000111010001000\cdots$

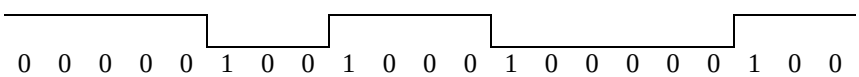
(**٧,٢,١٤**) لتكن $C_i = (n_i, k_i, d_i)$ حيث $i = 1,2$ شفرتين خطيتين. لنفرض أن C_1 ورقت بينياً مع C_2 للحصول على كلمات شفرة. جد طول الخطأ الاندفاعي الذي يمكن تصويبه باستخدام خوارزمية فك التشفير المقدمة في المثال (**٧,٢,١١**) إذا كانت كلمات الشفرة هذه:

(أ) ورقت بينياً لعمق s قبل إرسالها.

(ب) استُخْدِم مؤجل للتوريق البيني لعمق s قبل ارسالها.

(٧,٣) تطبيقات على الأقراص المدمجة
Applications to Compact Discs

أحدث استخدام الأقراص المدمجة لتسجيل الألحان الموسيقية تغييراً أساسياً في عالم الموسيقى، حيث إن السبب وراء الجودة العالية للألحان المسجلة على الأقراص المدمجة هو استخدام شفرات تصويب الأخطاء عند تخزين هذه الألحان. يتكون على القرص المدمج مسار حلزوني على نقرات أو ندب صغيرة (مستوى منخفض) ومن ثم يتبع ذلك حزمة من أشعة الليزر (أو اللازر) لتحديد التغيرات في ارتفاع المسار الحلزوني وذلك باكتشاف التغيرات في شدة كمية الضوء المنعكسة من القرص المدمج. ينتج عن ذلك كلمات ثنائية حيث يقابل كل تغير في الارتفاع الإحداثي 1 ويقابل عدم التغير في الارتفاع الإحداثي 0.



يتم أثناء عملية التسجيل أخذ 44100 عينة موسيقية في كل ثانية حيث يقابل سعة الموجة الصوتية لكل عينة كلمة ثنائية طولها 16. ولهذا يقسم مدى السعات إلى 2^{16} قيمة. تحتاج عملية التسجيل على **الستيريو (Stereo)** إلى قياسين للسعة يؤخذان 44100 مرة في كل ثانية، واحد من اليسار والآخر من اليمين.

لأغراض التشفير يتم تمثيل كل كلمة ثنائية طولها 16 التي تقابل قياس سعة بعنصرين في الحقل $GF(2^8)$ (يُطلق مصطلح بايت byte على كل عنصر من عناصر الحقل). أثناء عملية التسجيل يتم انتاج 4 بايتات هي m_{4t} ، m_{4t+1} ، m_{4t+2} ، m_{4t+3} عند كل **تكّة "tick"** t حيث قيمة التكّة تساوي $\frac{1}{44100}$ من الثانية. بعد ذلك يتم تجميع قياسات سعة من 6 تكّات متتالية $m_{24t}, m_{24t+1}, \cdots, m_{24t+23}$ للحصول على رسالة M_t طولها 24 حيث كل بايت ينتمي إلى $GF(2^8)$. لنفرض أن C هي الشفرة $RS(2^8, 5)$. عندئذ، يتم تشفير الرسالة M_t إلى كلمة شفرة c_t باستخدام الشفرة $C_1 = C(227)$ التي هي مقصور شفرة ريد وسولومن على الحقل $GF(2^8)$ حيث $(n_1, k_1, d_1) = (28,24,5)$ (انظر المثال **(٦,٢,١١)**).

بهذا نكون قد استخدمنا إطاراً مؤجلاً للتوريق البيني من النوع 4 لكلمات الشفرة التي حصلنا عليها (انظر الجدول **(٧,٢)**). لاحظ أن طول كل من أعمدة صفيف الجدول **(٧,٢)** في حالتنا هذه يساوي $n_1 = 28$. وبما أن البايتات في كلمة الشفرة c_t تقع في الأعمدة $t, t+4, t+8, \cdots, t+108$ فمن الطبيعي أن نرمز لهذه البايتات بالرموز $c_{1,t}, c_{2,t+4}, c_{3,t+8}, \cdots, c_{28,t+108}$.

يحتوي العمود t من صفيف الجدول **(٧,٢)** على البايتات $c_{1,t}, c_{2,t}, \cdots, c_{28,t}$ (تذكر أن $c_{i,j}$ هي البايت i في كلمة الشفرة $c_{j-4(i-1)}$)، وهذه قد استخدمت كرسائل من الطول 28 على الحقل $GF(2^8)$ ومن ثم شُفّرت باستخدام $C_2 = C(223)$ وهي شفرة ريد وسولومن المقصورة على $GF(2^8)$ حيث $(n_2, k_2, d_2) = (32,28,5)$.

يُضاف بايت لكل من كلمات الشفرة C_2 لغرض السيطرة والعرض ومن ثم يكون طول كلمات الشفرة يساوي 33.

جميع البايتات لحد الآن إما أنها تحمل معلومات وإما تم إضافتها لغرض اكتشاف وتصويب الأخطاء. ولكن يظهر عند التطبيق العملي لمسار الليزر أن التغيرات في ارتفاع المسار الحلزوني لا تقع قريبة جداً من بعضها بعضاً ولا بعيدة جداً بعضها عن بعض. ولهذا فقد تقرر أن يظهر على الأقل صفران وعلى الأكثر عشرة أصفار بين كل ظهورين متتاليين للواحد في التمثيل الثنائي لكلمة الشفرة. شفرة رید وسولومن لا تتمتع بهذه الخاصية ولكن يوجد 267 كلمة ثنائية طول كل منها 14 تتمتع بهذه الخاصية. يتم مقابلة عناصر الحقل وعددها 256 مع 256 من هذه الكلمات الثنائية (توضع عادة في جدول) وتهمل 11 كلمة ثنائية. تُسمى هذه العملية، تغييراً في طبقة الصوت من ثمانية إلى أربعة عشر (اختصاراً EFM). ولغرض التأكد من أن هذه الخاصية تتحقق بين الكلمات من الطول 14 يُضاف 3 بايتات أخرى (إما كلها 0 وإما كلها 1). وبهذا يكون طول كل تمثيل بياني لكلمة شفرة يساوي $561 = 17 \times 33$.

وأخيراً، يُضاف لغرض المزامنة كلمة ثنائية طولها 27 لكل كلمة شفرة بحيث تبقى الخاصية المقدمة في الفقرة السابقة محققة. وبهذا يتم بداية تخزين المعلومات الصوتية لست تكّات متتالية كمتجه ثنائي طوله $196 = 8 \times 24$ وبعد إتمام جميع العمليات يظهر على القرص المدمج ككلمة ثنائية طولها 588.

يبقى علينا مناقشة فك التشفير. نقوم أولاً بمعالجة الخطوات الزائدة عكسياً مثل EFM على أمل أن تكون الكلمات المستقبلة هي كلمات شفرة تنتمي إلى C_1 (انظر الملحوظة في نهاية هذا البند). تستخدم الشفرة C_2 لتصويب خطأ واحد في جميع الكلمات. وإذا تم اكتشاف أكثر من خطأ فنقوم بتعليم جميع بايتات الكلمة المستقبلة (انظر البند **٧,٢**) للتوريق البيني بين شفرتين). وبهذا يتم التخلص من تأثير الاطار

المؤجل للتوريق البيني من النوع 4. وأخيراً تستخدم الشفرة C_1 لتصويب أخطاء لا يزيد عددها عن 4 أخطاء (تذكر أن مسافة C_1 تساوي 5) على اعتبار أن جميع البايتات المعلّمة أخطاء والبايتات غير المعلّمة هي إحداثيات صحيحة.

هل فك التشفير هذا جيد؟ للاجابة عن ذلك، لاحظ أولاً أن الشرط الوحيد الذي ينتج عنه خطأ في استخدام C_2 لفك التشفير هو أن تكون المسافة بين الكلمة المستقبلة وكلمة شفرة تنتمي إلى C_2 ولكنها ليست الكلمة المرسلة لا تزيد عن 1. ولكن عدد أنماط الأخطاء التي تتمتع بهذه الخاصية قليل جداً؛ لأن عدد كلمات الشفرة C_2 هو $(2^r)^k = (2^8)^{28} = 2^{224}$ وواحدة فقط من هذه الكلمات هي الكلمة المنشودة وجميع الكلمات المتبقية وعددها $2^{224} - 1$ تقع على مسافة 1 من كلمات عددها $1 + 32(2^8 - 1)$ وطول كل منها يساوي 32. وبهذا نرى أنه من بين جميع أنماط الأخطاء الثنائية التي عددها $(2^8)^{32}$ المحتمل إضافتها إلى كلمة من كلمات الشفرة C_2 يوجد من بينها فقط عدد $(2^{224} - 1)(1 + 32(2^8 - 1))$ كلمة تقع على مسافة مقدارها 1 من كلمة أخرى من كلمات الشفرة C_2. أي أن هذا العدد هو تقريباً $1/2^{19}$ من هذه الكلمات. تم تصميم هذا الاستخدام للشفرة C_2 التي تصوّب نمط خطأ واحد لمعالجة الأخطاء العشوائية الصغيرة التي تحدث أثناء طلاء الأقراص المدمجة وقطعها.

أيضاً، بعد إزالة تأثير الاطار المؤجل للتوريق البيني من النوع 4 يتم فك تشفير الكلمة المستقبلة إلى كلمة شفرة صحيحة من كلمات C_1 إذا كان عدد الإحداثيات المعلّمة في الكلمة لا يزيد عن 4 (بافتراض أن C_2 تكتشف جميع الأخطاء وهذا هو الوضع في غالب الأحيان كما بيّنا سابقاً). ولكن قبل تأثير اندفاع واحد على 5 إحداثيات من كلمة في الشفرة C_1 يجب أن يؤثر على 17 عموداً من صفيف الجدول (**٧,٢**). أي على $2 + 2 + 32 \times 15$ بايتاً على الأقل (إذا تغير بايتان من العمود الأول أو العمود السابع عشر فينتج عن ذلك تعليم جميع بايتات هذا العمود بواسطة C_2).

وبما أن كل من أعمدة الجدول (**٧,٢**) يقابل كلمة طولها 588 على القرص المدمج فنرى أن جميع الاندفاعات من الطول $8871 = 17 \times 3 + 588 \times 15$ يتم فك تشفيرها بصورة صحيحة. يقابل هذا الطول الاندفاعي ما يقارب من $2.5mm$ من طول مسار على القرص المدمج.

ملحوظة

لاحظ أننا قمنا بتوضيح أوجه عملية التشفير المهمة فقط حيث توجد عمليات توريق بيني أخرى عند التطبيق العملي. على سبيل المثال، يتم إزاحة جميع البايتات التي تقع في مواقع فردية من كلمات الشفرة C_2 مواقع عددها $n_2 = 32$ بحيث يتم خلطها مع البايتات ذات المواقع الزوجية في كلمة الشفرة التالية مما يحسن من فرص قدرة الشفرة C_2 من تصويب نمط خطأ واحد؛ وذلك لأن خطأين متتاليين يؤثران الآن على كلمتي شفرة مختلفتين.

أيضاً، يتم إعادة ترتيب البايتات في كلمات الشفرة C_1. كل من هذه الكلمات يحتوي معلومات 6 تكّات متتالية من اليسار واليمين ولتكن $L_1, L_2, \cdots, L_6$ و $R_1, R_2, \cdots, R_6$ إضافة إلى رمزين للنوعية Q_1 و Q_2 يتم اضافتهما عند استخدام C_1 للتشفير. يتم ترتيب ذلك على النحو التالي:

$$L_1 L_3 L_5 R_1 R_3 R_5 Q_1 Q_2 L_2 L_4 L_6 R_2 R_4 R_6$$

الغرض من ذلك هو أنه لو بقي عدد من البايتات المتتالية معلّماً بعد عملية فك التشفير فتعامل على أنها معلومات غير موثوقة. وفي هذه الحالة يمكن استبدال قيمة غير موثوق بها L_i بالسعة التي وجدت باستكمال القيمتين الموثوقتين L_{i-1} و L_{i+1}. على سبيل المثال، إذا بقيت القيم $L_1, L_5, R_1, R_3, R_5, Q_1, Q_2$ معلّمة فنستطيع إيجاد القيمة L_3 كوسط حسابي لسعتي القيمتين الموثوقتين L_2 و L_4 وهكذا.