الفصل الأول

مقدمة في نظرية التشفير
Introduction to Coding Theory

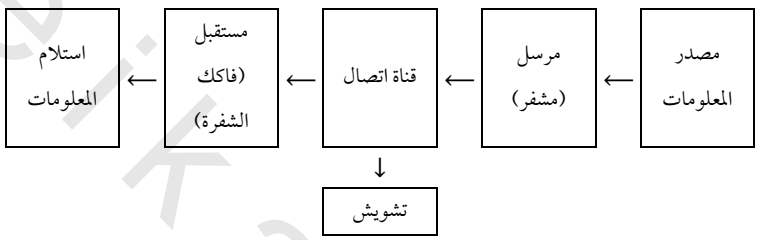
(١,١) مقدمة
Introduction

نظرية التشفير (أو الترميز) هي دراسة طرائق النقل الفعّال والدقيق للمعلومات من مكان إلى آخر. تم تطوير هذه النظرية لتغطية تطبيقات متنوعة مثل خفض التشويش في التسجيل على الأقراص المدمجة إلى الحد الأدنى، عملية إرسال المعلومات المالية عبر خطوط الهاتف، عملية نقل البيانات من جهاز حاسب لآخر أو من الذاكرة إلى المعالج المركزي، إرسال المعلومات من مصادر بعيدة مثل الأقمار الصناعية الجويّة أو أقمار الاتصالات الصناعية أو مركبة فضائية تستخدم لإرسال صور من كوكبي المشتري وزحل إلى الكرة الأرضية.

يُسمى الوسط المادي الذي يتم نقل المعلومات بواسطته، قناة اتصال أو **قناة (Channel)**. خطوط الهاتف والغلاف الجوي مثالان على القناة. يُسمى الإزعاج غير المرغوب فيه الذي ينتج عنه اختلاف بين المعلومات المرسلة والمعلومات المستقبلة، **التشويش (Noise)**. إن مسببات التشويش عديدة مثل: كُلَف الشمس (بقع داكنة تبدو بين فترة وأخرى على سطح الشمس)، البرق، انحناءات في شريط مغناطيسي، أمطار،

تداخلات في خطوط الهاتف، إزعاج عشوائي في المذياع، أخطاء مطبعية، ضعف في السمع، عدم وضوح في الكلام وأمثلة أخرى كثيرة.

ينصب اهتمام نظرية التشفير على مسألة اكتشاف وتصويب أخطاء الارسال الناتجة عن التشويش في قناة الاتصال. المخطط التالي يقدم لنا فكرة عامة عن ماهية نظام إرسال معلومات.



إن أهم جزء من هذا المخطط بالنسبة لنا هو التشويش؛ لأنه بغياب التشويش تكون نظرية التشفير عديمة الفائدة.

في التطبيق العملي يكون بمقدورنا تقليل التشويش عند اختيارنا لقناة اتصال مناسبة لنقل المعلومات واستخدام مرشحات تشويش مختلفة لمقاومة بعض أنماط التدخلات التي يمكن أن تقابلنا وهذا من مهام المهندسين. وبمجرد الاتفاق على اختيار أفضل نظام ميكانيكي لحل هذه المسائل نستطيع التركيز على إنشاء **المشفّر وفاكك التشفير (Encoder & Decoder)** وتكون رغبتنا في هذا الإنشاء تحقيق ما يلي:

(١) سرعة تشفير المعلومات.

(٢) سهولة نقل الرسائل المشفّرة.

(٣) سرعة فك تشفير الرسائل المستقبلة.

(٤) تصويب الأخطاء الناتجة عن التشويش.

(٥) القدرة على نقل معلومات بحد أقصى لكل وحدة زمن.

الخاصية (٤) هي الهدف الأساس لنظرية التشفير، ولكن هذه الخاصية ليست متناغمة مع الخاصية الخامسة ومن الممكن أن لا تكون متناغمة مع باقي الخواص. ولذا لإيجاد حل فلا بد من المقايضة بين الأهداف الخمسة.

نستخدم في اتصالاتنا اليومية كلمات سواء أكانت شفهية أم مكتوبة وهذه الكلمات مكوّنة من حروف هجائية محدودة. ولتبادل معلومات نقوم بتشفيرها إلى متتالية من الكلمات ومن ثم نتكلم هذه الكلمات أو نكتبها وبعد ذلك نرسلها عبر قناة اتصال وهي في العادة الفضاء من الفم إلى الأُذن أو من القلم إلى الورقة ومن ثم إلى العين. أما التشويش فقد يتسبب من عدم وضوح في الكلام أو ضعف في السمع أو خطأ نحوي أو موسيقى عالية أو تداخل في الكلام أو خطأ في الإملاء أو خطأ في القراءة أو خطأ مطبعي. وأخيراً يكون فك التشفير هو قراءتنا (أو سماعنا) وفهمنا للرسالة المستقبلة. ونتيجة لذلك نكون قد أنشأنا أدوات لتصويب الأخطاء دون أن نتعمد ذلك. فلنفرض أننا استقبلنا الرسالة "Apt natural. I have a gub" وهي تحذير مكتوب لعملية سطو مأخوذ من فيلم "احصل على النقود واهرب" **لودي آلن (Woody Allen)**. وبما أن كلمات الهجائية الإنجليزية ذات الطول الواحد والتي لها معنى هي كلمات محدودة فيكون من الواضح أن "gub" ليست كلمة ذات معنى. وبهذا فإننا نفترض أن الكلمة المرسلة قريبة من الكلمة "gub". ومن ثم فهي على الأرجح "gut" أو "gun" أو "tub" وليست "firetruck" أو "rat". ومن فحوى الرسالة فقط نرجح أن الكلمة هي "gun". أما الكلمة "Apt" فهي كلمة ذات معنى (تعني ملائم) ولكنها لا تتلاءم مع فحوى الرسالة ومن ثم نرجح وقوع خطأ في الإرسال ونُصوّبها على أنها "act". وإذا كنا متعلمين ومتقنين لقواعد اللغة فإننا نقوم بتصويب "natural" لتكون "naturally" على الرغم من أن الاحتمال الأكبر لهذا الخطأ هو المصدر وليس التشويش في قناة الاتصال. سنتعامل فقط مع الأخطاء من النوع الأول. أي سنختار الكلمة التي على الأرجح قد تم إرسالها.

إن الطريقة التقليدية المتبعة لتجنب الأخطاء هي تذييل الرسالة بمعلومات زائدة حيث عديد من الجهات تضيف رقماً إضافياً إلى الأعداد المستخدمة للتعريف بالمنتج وتستخدم هذه الإضافات لاختبار صحة البيانات أو أرقام الحسابات وهذه هي الطريقة الشائعة الاستخدام في عملية التشفير في الأعمال اليومية. الأفكار التي سنناقشها في هذا الكتاب هي أفكار مشابهة ولكنها أكثر تطوراً.

(١,٢) فرضيات أساسية
Basic Assumptions

نقدم الآن بعض التعاريف والفرضيات الأساسية التي سنستخدمها في هذا الكتاب.

في عديد من الحالات تُستخدم المتتاليات الثنائية (حدودها مكوّنة من الرقمين 0 و 1) لنقل المعلومات المزمع إرسالها. ولهذا يُسمى كل من الرقمين 0 أو 1 **إحداثياً (Digit)**. **الكلمة (Word)** هي متتالية من الإحداثيات. **طول الكلمة (Length of Word)** هو عدد الإحداثيات المكوّنة للكلمة. فمثلاً 0110101 كلمة طولها سبعة. وتتم عملية نقل الكلمة بإرسال الإحداثيات واحدة بعد الأخرى عبر **قناة ثنائية (Binary Channel)**. وقناة ثنائية هنا تعني أن الإحداثيين المستخدمين هنا هما 0 و 1 فقط. كل إحداثي من إحداثيات الكلمة يتم إرساله ميكانيكياً أو كهربائياً أو مغناطيسياً أو بأي وسيلة أخرى، وأياً كانت الوسيلة فذلك يكون على شكل نبضات يسهل تمييز بعضها عن بعض (أي أن نبضة الإحداثي 0 مختلفة عن نبضة الإحداثي 1).

تعرّف **الشفرة الثنائية (Binary Code)** على أنها مجموعة C من الكلمات. فمثلاً، الشفرة المكوّنة من جميع الكلمات ذات الطول 2 هي:

$$C = \{00, 10, 01, 11\}$$

الشفرة القالبية (Block Code) هي شفرة أطوال جميع كلماتها متساوية ويُسمى هذا الطول الموحّد، **طول الشفرة (Length of the Code)**. سندرس في هذا الكتاب

الشفرات القالبية فقط ولهذا فعند قولنا شفرة نعني دائماً أنها شفرة قالبية ثنائية. تُسمى الكلمات المكوّنة لشفرة معطاة C، **كلمات شفرة (Code Words)**. سنرمز لعدد كلمات شفرة C بالرمز $|C|$.

تمارين

(**١,٢,١**) اكتب جميع الكلمات ذات الطول 3 وجميع الكلمات ذات الطول 4 وجميع الكلمات ذات الطول 5.

(**١,٢,٢**) جد صيغة لعدد الكلمات من الطول n

(**١,٢,٣**) لتكن C الشفرة المكوّنة من جميع الكلمات ذوات الطول 6 بحيث تحتوي كل كلمة على عدد زوجي من الإحداثيات 1. اكتب كلمات الشفرة C.

نحتاج أيضاً لوضع بعض الفرضيات الأساسية على قناة الاتصال، هذه الفرضيات ستحدد الشكل الذي تقوم عليه نظرية التشفير.

الفرضية الأولى هي افتراض أن كلمة الشفرة من الطول n مكوّنة من إحداثيات 0 وإحداثيات 1 ويتم استقبالها ككلمة طولها n مكوّنة من إحداثيات 0 وإحداثيات 1 والتي لا تكون بالضرورة الكلمة المرسلة نفسها.

الفرضية الثانية هي سهولة تحديد بداية كل من الكلمات المرسلة. على سبيل المثال، عند استخدامنا لكلمات شفرة من الطول 3 واستقبالنا للمتتالية 011011001 فتكون الكلمات المرسلة هي 011 ,011 ,001 على التوالي (تتم قراءة المتتالية من اليسار إلى اليمين)، وهذا يعني استحالة أن تكون القناة قد أرسلت المتتالية 01101100 إلى المستقبل؛ لأن عدد الإحداثيات لا يقسم على 3.

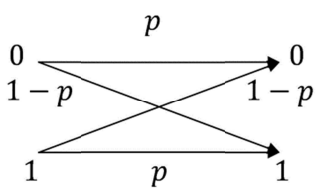
والفرضية الأخيرة هي أن التشويش متناثر عشوائياً وليس على شكل كتل تُدعى **اندفاعات (Bursts)**. إن هذا يعني أن احتمال تأثّر إحداثي أثناء الإرسال يساوي احتمال تأثر أي إحداثي آخر ولا يتأثر بالأخطاء التي قد تكون حصلت لإحداثيات

مجاورة. هذه الفرضية ليست واقعية للعديد من أنماط التشويش مثل البرق أو خدوش الأقراص المدمجة. سنتعامل لاحقاً مع مثل هذه الأنماط من التشويش.

في حالة القناة المثالية (عدم وجود تشويش)، يكون الإحداثي المرسل 0 أو 1 هو الإحداثي المستقبل. وبهذا إذا كانت جميع القنوات مثالية فلا حاجة لنا إلى نظرية التشفير. ولكن لحسن الحظ (أو ربما لسوء الحظ) جميع القنوات غير مثالية ومشوّشة ولكن تشويش بعضها أقل من تشويش بعضها الآخر. أي أن بعضها أكثر موثوقية من بعضها الآخر.

نقول إن القناة الثنائية **متماثلة (Symmetric)** إذا كانت الدقّة في ارسال 0 و 1 متساوية. أي أن احتمال استقبال الإحداثي الصحيح لا يعتمد على أي من الإحداثيين 0 أو 1 قد يكون أُرسل. وتعرّف **موثوقية (Reliability)** قناة ثنائية متماثلة (اختصاراً BSC) على أنها عدد حقيقي p حيث $0 \leq p \leq 1$ و p هو احتمال أن يكون الإحداثي المرسل هو الإحداثي المستقبل (أي احتمال عدم حدوث خطأ في الإحداثي).

ولهذا إذا كان p هو احتمال أن يكون الإحداثي المستقبل هو نفس الإحداثي المرسل فإن $1 - p$ هو احتمال أن يكون الإحداثي المستقبل ليس هو الإحداثي المرسل. المخطط التالي يبين عمل القناة BSC:



في معظم الأحيان يكون من الصعب حساب الموثوقية p بصورة دقيقة لقناة معطاة ولكن ذلك ليس له تأثير مباشر على نظرية التشفير.

نقول إن قناة ما أكثر موثوقية من قناة أخرى إذا كانت موثوقيتها أعلى من موثوقية القناة الأخرى. لاحظ أن الحالة $p = 1$ لا تهمنا؛ لأنه لا يوجد أخطاء في الإرسال ومن ثم فإن القناة مثالية. كما أن القنوات التي يكون فيها $p = 0$ ليست بذات أهمية. لاحظ أيضاً إمكانية تحويل قناة تحقق $0 < p \leq \frac{1}{2}$ إلى قناة تحقق $\frac{1}{2} \leq p < 1$. ولهذا فإننا سنفترض دائماً أن قنوات BSC المستخدمة هي قنوات باحتمال p حيث $\frac{1}{2} < p < 1$. (الحالة التي يكون فيها $p = \frac{1}{2}$ هي فحوى التمرين (١,٢,٦)).

تمارين

(١,٢,٤) بيّن لماذا تكون القناة ليست بذات أهمية عندما $p = 0$.

(١,٢,٥) بيّن كيفية تحويل قناة تحقق $0 < p \leq \frac{1}{2}$ إلى قناة تحقق $\frac{1}{2} \leq p < 1$.

(١,٢,٦) ماذا يمكن القول عن قناة تحقق $p = \frac{1}{2}$ ؟

(١,٣) تصويب واكتشاف أنماط الأخطاء

Correcting & Detecting Error Patterns

ندرس الآن امكانية تصويب واكتشاف الأخطاء. ففي هذا البند نعالج مفهومي تصويب واكتشاف الأخطاء حدسياً ونؤجل المعالجة الرياضية لبنود لاحقة.

لنفرض أننا استقبلنا كلمة ووجدنا أنها ليست كلمة شفرة. عندئذ، يكون من الواضح أن خطأ ما قد حصل أثناء عملية الإرسال وبهذا نكون قد اكتشفنا خطأ (أو عدة أخطاء). أما إذا كانت الكلمة المستقبلة هي كلمة شفرة فتكون إمكانية عدم حصول أخطاء في الإرسال واردة ومن ثم لا نستطيع اكتشاف أي خطأ. أما مفهوم تصويب خطأ فيحتاج إلى المزيد من التمحيص. وكما بيّنا في المقدمة عند ميلنا لتصويب الكلمة "gub" إلى "gun" وليس إلى "rat" فحدسُنا يقودنا إلى اقتراح تصويب كلمة مرسلة إلى كلمة شفرة بحيث تكون الأقرب إلى الكلمة المرسلة (أي تغيير أقل عدد ممكن من

الإحداثيات). سنبيّن في بند لاحق أن احتمال أن تكون كلمة الشفرة هذه هي الكلمة التي تم إرسالها لا يقل عن احتمال إرسال أي كلمة شفرة أخرى. سنوضح ذلك بدراسة بعض الأمثلة للشفرات مع ملاحظة افتراضنا عدم سقوط أو إضافة أي إحداثي أثناء الإرسال. فمثلاً، نفترض استحالة فك تشفير "gub" إلى "firetruck".

مثال (١,٣,١)

لنفرض أن $C_1 = \{00, 01, 10, 11\}$. عندئذ، جميع الكلمات المستقبلة هي كلمات شفرة ومن ثم فإن C_1 لا تستطيع اكتشاف أي خطأ. كما أن C_1 لا تصوّب أي أخطاء؛ لأن جميع الكلمات المستقبلة هي كلمات شفرة ومن ثم لا تتطلب أي تغيير فيها. ▲

مثال (١,٣,٢)

إذا كرّرنا كلاً من كلمات الشفرة C_1 ثلاث مرات نحصل على الشفرة:

$$C_2 = \{000000, 010101, 101010, 111111\}$$

هذا مثال على ما يُسمى **شفرة مكرّرة (Repetition Code)**. لنفرض أننا استقبلنا الكلمة 110101. بما أن هذه ليست كلمة شفرة، فلذا لا بد من وقوع خطأ واحد على الأقل أثناء عملية الإرسال. وبملاحظة أنه يمكن الحصول على كلمة الشفرة 010101 بتغيير إحداثي واحد فقط ولكن يتطلب الحصول على كلمات الشفرة الأخرى إلى تغيير أكثر من إحداثي واحد، فنرجح أن تكون كلمة الشفرة 010101 هي المرسلة وبهذا نُصوّب 110101 إلى 010101. (تُسمى كلمة الشفرة التي نحصل عليها من كلمة w بتغيير أقل عدد من الإحداثيات، كلمة **الشفرة الأقرب (Closest Codeword)** وسنقوم لاحقاً بتعريفها بدقة أكثر). في الحقيقة، إذا تم إرسال أي كلمة شفرة $c \epsilon C_2$ ووقع خطأ واحد أثناء الإرسال فإن كلمة الشفرة الوحيدة الأقرب إلى الكلمة المستقبلة هي c نفسها ومن ثم فإن الشفرة C_2 تصوّب خطأ واحد فقط. ▲

مثال (١,٣,٣)

إذا أضفنا إحداثياً ثالثاً لكل كلمة من كلمات الشفرة C_1 بحيث يصبح عدد الإحداثيات 1 في كل من كلمات الشفرة زوجياً نحصل على الشفرة:

$$C_3 = \{000\,,011\,,101\,,110\}$$

يسمى الإحداثي المضاف **إحداثي اختبار النوعية (Parity–Check Digit)**. إذا استقبلنا الكلمة 010 وهي ليست كلمة شفرة يكون بإمكاننا اكتشاف وقوع خطأ في الإرسال ويمكن الحصول على كل من كلمات الشفرة 110 و 000 و 11 بتغيير إحداثي واحد في الكلمة المستقبلة. في البنود القادمة سنفرق بين تعاملنا مع الكلمات المستقبلة الأقرب إلى كلمة شفرة وحيدة (ومن ثم تكون كلمة الشفرة الوحيدة المرجح إرسالها) كما هو الحال في المثال (١,٣,٢) وبين الكلمات المستقبلة الأقرب إلى عديد من كلمات الشفرة كما هو الحال في هذا المثال. سنكتفي في هذه المرحلة بملاحظة أنه من الأنسب تصويب 010 لتكون أي من 110، 000، 011 ولكن ليس 101. ▲

تمارين

(١,٣,٤) لتكن C شفرة جميع الكلمات ذات الطول 3. إذا استقبلنا الكلمة 001 فبيّن أي كلمة شفرة تكون قد أُرسلت على الأرجح.

(١,٣,٥) أضف إحداثي اختبار النوعية لكلمات الشفرة المقدمة في التمرين (١,٣,٤) ومن ثم استخدم الشفرة C الناتجة عن ذلك للإجابة عن الأسئلة التالية:

(أ) إذا استقبلنا الكلمة 1101 فهل بإمكاننا اكتشاف خطأ ؟

(ب) إذا استقبلنا الكلمة 1101 فما هي كلمة الشفرة التي تكون على الأرجح قد أُرسلت ؟

(ج) هل توجد كلمة طولها 4 وليست كلمة شفرة بحيث تكون الأقرب إلى كلمة شفرة وحيدة ؟

(١,٣,٦) كرّر كلاً من كلمات الشفرة C المبينة في التمرين **(١,٣,٤)** ثلاث مرات لتحصل على شفرة تكرار من الطول 9. جد كلمة الشفرة الأقرب إلى كل من الكلمات المستقبلة:

(أ) 001000001 (ب) 011001011

(ج) 101000101 (د) 100000010

(١,٣,٧) جد أكبر عدد من كلمات الشفرة ذات الطول 4 المحتواة في شفرة تستطيع اكتشاف خطأ واحد أياً كان هذا الخطأ.

(١,٣,٨) كرّر التمرين **(١,٣,٧)** عندما يكون $n = 5$ و $n = 6$ وبعد ذلك لأي n.

(١,٤) معدل المعلومات
Information Rate

يتضح لنا من البند السابق أن إضافة إحداثيات لكلمات الشفرة يزيد من قدرة الشفرة على تصويب واكتشاف الأخطاء. ومن الواضح أيضاً أنه كلما ازداد طول كلمة الشفرة ازداد الزمن اللازم لإرسال الرسالة. **معدل المعلومات (Information Rate)** للشفرة هو عدد مصمم لقياس الجزء من كل كلمة شفرة الذي يتضمن الرسالة. يعرّف معدل معلومات الشفرة C ذات الطول n (للشفرات الثنائية) على أنه:

$$\frac{1}{n} log_2|C|$$

وبما أن $1 \leq |C| \leq 2^n$ فمن الواضح أن معدل المعلومات يقع بين 0 و 1. فهو يساوي 1 إذا كانت كل كلمة هي كلمة شفرة ويساوي 0 إذا كان $|C| = 1$.

على سبيل المثال، معدل معلومات الشفرات C_1 ، C_2 ، C_3 المقدمة في البند السابق هو 1، $\frac{1}{3}$ ، $\frac{2}{3}$ على التوالي. كل من معدلات المعلومات هذه تتلاءم مع الشفرة ذات الصلة. لاحظ أن أول إحداثيين من الستة إحداثيات لكل من كلمات الشفرة C_2

هما اللذان يتضمنان الرسالة وأن الإحداثيين الأول والثاني من الإحداثيات الثلاثة لكل من كلمات الشفرة C_3 هما اللذان يتضمنان الرسالة.

تمرين

(١,٤,١) احسب معدل المعلومات لكل من الشفرات المقدمة في التمارين **(١,٣,٤)**، **(١,٣,٥)**، **(١,٣,٦)**.

(١,٥) تأثير تصويب واكتشاف الأخطاء

The Effect of Error Correction & Detection

لفهم مدى التأثير الناتج عن إضافة إحداثي اختبار النوعية إلى شفرة على اكتشاف وتصويب الأخطاء، ندرس المثال التالي:

نفرض أن جميع الكلمات ذات الطول 11 وعددها $2^{11} = 2048$ هي كلمات شفرة. عندئذ، لا يمكننا اكتشاف أي خطأ. لنفرض أن موثوقية القناة هي $p = 1 - 10^{-8}$ ولنفرض أن معدل إرسال الإحداثيات هو 10^7 إحداثي في الثانية. حينئذ، يكون احتمال وجود خطأ في الكلمة المرسلة هو $11p^{10}(1-p) \approx 11/10^8$. وبهذا نرى أن عدد الكلمات المرسلة والتي تحتوي على خطأ لا يتم اكتشافه هو $\frac{11}{10^8} \times \frac{10^7}{11} = 0.1$ كلمة في الثانية الواحدة أو كلمة واحدة خطأ كل 10 ثوانٍ أو 6 كلمات خطأ كل دقيقة أو 360 كلمة خطأ كل ساعة أو 8640 كلمة خطأ كل يوم !

لنفرض الآن أننا أضفنا إحداثي اختبار النوعية لكل كلمة من كلمات الشفرة بحيث يكون عدد الإحداثيات 1 في كل منها زوجياً. عندئذ، يمكن اكتشاف أي خطأ واحد ومن ثم لكي تحتوي الكلمة المرسلة على أخطاء لا يمكن اكتشافها فيجب أن يكون عدد الأخطاء على الأقل 2. ولذا فاحتمال وقوع خطأين على الأقل يساوي:

$$\binom{12}{2} P^{10}(1-p)^2 = 66(1-10^{-8})^{10} \times 10^{-16} \approx 66 \times 10^{-16}$$

ويكون عدد الكلمات المرسلة والتي تحتوي أخطاء لا يتم اكتشافها هو $\frac{66}{10^{16}} \times \frac{10^7}{12} \approx 5.5 \times 10^{-9}$ كلمة في الثانية أو كلمة واحدة كل 2000 يوم !

مما سبق نرى أن إضافة إحداثي واحد إلى كل من كلمات الشفرة ليصبح الطول يساوي 12 عوضاً عن 11 يُبيّن لنا سهولة اكتشاف الأخطاء عند وقوعها. ولتصويب هذه الأخطاء نطلب إعادة إرسال الرسالة وهذا يعني توقف عملية الإرسال حتى نحصل على تأكيد أو تخزين الرسائل لفترة مؤقتة لحين طلب إعادة الإرسال وتكون التكاليف باهظة في كلتا الحالتين سواء على صعيد الزمن اللازم أو سعة التخزين. كما أنه من الممكن أن تكون عملية إعادة الإرسال غير عملية كما في المراكب الفضائية أو استخدام الأقراص المدمجة. ولتخفيض الثمن الباهظ الناتج عن زيادة طول كلمات الشفرة فيكون من المناسب إضافة بعض قدرات تصويب الأخطاء للشفرة. ولكن إضافة مثل هذه القدرات قد ينتج عنه صعوبة في التشفير وفك التشفير ولكنه يساعد على تخفيض التكاليف في الزمن وسعة التخزين المبينة للشفرة المقدمة في بداية هذا البند.

إحدى الخطط المساعدة على تصويب الأخطاء هي إنشاء شفرة تكرار حيث يتم إرسال كل من كلمات الشفرة ثلاث مرات متتالية. فإذا وقع خطأ واحد على الأكثر في كل كلمة شفرة من الطول 33 فنكون قد ضمنا صحة إرسالين على الأقل من الإرسالات الثلاثة. وبما أن مقارنة ثلاث كلمات من الطول 11 أمر سهل فإن الثمن الوحيد الذي يدفع لتصويب خطأ هو تخفيض معدل المعلومات من 1 إلى $\frac{1}{3}$.

سنبين لاحقاً إمكانية إضافة 4 إحداثيات فقط لكل من كلمات الشفرة ذات الطول 11 لجعلها قادرة على تصويب خطأ واحد ومن ثم يكون معدل المعلومات هو $\frac{11}{15}$ وهو أفضل بكثير من $\frac{1}{3}$ إذا تجاهلنا الإعاقة الناتجة عن ذلك في عملية التشفير وفك التشفير.

نخلص إلى القول إن مهمتنا هي تصميم شفرات بمعدل معلومات معقول وثمن معقول لعملتي التشفير وفك التشفير مع القدرة على تصويب واكتشاف بعض الأخطاء لتلافي الحاجة إلى إعادة الإرسال.

(١,٦) إيجاد الاحتمالية القصوى لكلمة الشفرة المرسلة
Finding the Most Likely Codeword Transmitted

لنفرض أن لدينا فكرة عامة عن عملية الإرسال وأننا على معرفة بكلمة الشفرة v المرسلة والكلمة w المستقبلة. لكل v و w نفرض أن $\varphi_p(v,w)$ هو احتمال استقبال الكلمة w إذا كانت v هي كلمة الشفرة المرسلة عبر قناة BSC بموثوقية p. وبما أننا افترضنا أن توزيع التشويش هو توزيع عشوائي نرى أن كل إرسال إحداثي هو **حدث (Event)** مُستقل. وبهذا نرى أنه إذا اختلفت الكلمتان v و w في عدد d من المواقع فيكون عدد الإحداثيات المرسلة بدون أخطاء يساوي $n-d$ وعدد الإحداثيات المرسلة بأخطاء يساوي d ونحصل على:

$$\varphi_p(v,w) = p^{n-d}(1-p)^d$$

مثال (١,٦,١)

لتكن C شفرة من الطول 5 ولتكن $v \in C$. حينئذ، احتمال استقبال v دون أخطاء هو:

$$\varphi_p(v,v) = p^5$$

وإذا كانت $v = 10101 \in C$ وكانت $w = 01101$ و $p = 0.9$ فإن:

$$\varphi_p(v,w) = \varphi_p(10101,01101) = p^3(1-p)^2 = (0.9)^3(0.1)^2 = 0.00729.$$

▲

تمرين

(١,٦,٢) احسب $\varphi_{0.97}(v,w)$ لكل زوج من الكلمات v و w فيما يلي:

(أ) $v = 01101101\ , w = 10001110$

(ب) $v = 1110101\ , w = 1110101$

(ج) $v = 00101\ , w = 11010$

(د) $v = 00000\ , w = 00000$

(هـ) $v = 1011010\ , w = 0000010$

(و) $w = 01001 , v = 10110$

(ز) $w = 000010 , v = 111101$.

في التطبيق العملي نكون على معرفة بالكلمة المستقبلة w ولكننا لا نعرف كلمة الشفرة المرسلة v. كما نعلم أن كل كلمة شفرة v تُزودنا بتعيين للاحتمالات $\varphi_p(v,w)$ للكلمات المستقبلة w. كل من هذه التعينات هو نموذج رياضي وبهذا نختار النموذج (أي كلمة الشفرة v) التي تتفق مع معظم المشاهدات (في الحالة المعينة). أي نختار كلمة الشفرة التي لها احتمالية قصوى لكي تكون هي الكلمة المرسلة. وبهذا نفترض أن كلمة الشفرة v هي المرسلة عند استقبالنا للكلمة w إذا تحقق ما يلي:

$$\varphi_p(v,w) = max\{\varphi_p(u,w) : u \in C\}$$

تُزودنا المبرهنة التالية بمعيار لإيجاد كلمة الشفرة v.

مبرهنة (١,٦,٣)

لنفرض أن لدينا قناة BSC تحقق $\frac{1}{2} < p < 1$. لنفرض أن كلاً من v_1 و v_2 كلمة شفرة من الطول n وأن w كلمة طولها n. ولنفرض أن v_1 تختلف عن w بعدد d_1 من المواقع وأن v_2 تختلف عن w بعدد d_2 من المواقع. عندئذ:

$$d_2 \le d_1 \Leftrightarrow \varphi_p(v_1,w) \le \varphi_p(v_2,w)$$

البرهان

لاحظ أن

$$\varphi_p(v_1,w) \le \varphi_p(v_2,w) \Leftrightarrow p^{n-d_1}(1-p)^{d_1} \le p^{n-d_2}(1-p)^{d_2}$$

$$\Leftrightarrow \frac{p^{n-d_1}(1-p)^{d_1}}{p^{n-d_2}(1-p)^{d_2}} \le 1$$

$$\Leftrightarrow \left(\frac{p}{1-p}\right)^{d_2-d_1} \le 1$$

$$\Leftrightarrow d_2 \le d_1$$

لأن: $\frac{p}{1-p} > 1$. ■

وبهذا نكون قد أثبتنا وجود طريقة علمية لتصويب الأخطاء ويتم ذلك على النحو التالي: إذا استقبلنا الكلمة w فإننا نقوم بتصويب w إلى كلمة الشفرة التي تختلف عنها بأقل عدد من الإحداثيات؛ لأنه غالباً ما تكون كلمة الشفرة هذه هي المرسلة.

مثال (١,٦,٤)

لنفرض أن $w = 00110$ هي الكلمة المستقبلة عبر قناة BSC حيث $p = 0.98$. بيّن أي من كلمات الشفرة التالية تكون على الأرجح قد أُرسلت:

01101 ، 01001 ، 10100 ، 10101.

الحل

v	d (عدد مواقع الاختلاف مع w)
01101	3
01001	4
10100	2 ←
10101	3

استناداً إلى الجدول السابق والمبرهنة (١,٦,٣) نجد أن كلمة الشفرة التي تكون على الأرجح قد أُرسلت هي 10100. لاحظ أننا لا نحتاج لمعرفة قيمة p الدقيقة لاستخدام المبرهنة (١,٦,٣) ولكننا فقط بحاجة لمعرفة أن $p > \frac{1}{2}$. ▲

تمارين

(١,٦,٥) لنفرض أن $w = 0010110$ هي الكلمة المستقبلة عبر قناة BSC بموثوقية $p = 0.9$. أي من كلمات الشفرة التالية تكون على الأرجح قد أُرسلت:

1101001، 0011001، 0001110، 1111100، 1001011.

(١,٦,٦) أي من كلمات الشفرة الثمانية المبينة في التمرين (١,٣,٦) هي التي تكون على الأرجح أُرسلت إذا كانت الكلمة المستقبلة هي $w = 101000101$ ؟

(**١,٦,٧**) إذا كانت $C = \{01000, 01001, 00011, 11001\}$ وكانت $w = 10110$ هي الكلمة المستقبلة فما هي كلمة الشفرة التي تكون على الأرجح قد أُرسلت؟

(**١,٦,٨**) أعد التمرين (**١,٦,٧**) إذا كانت:

$$C = \{010101, 110110, 101101, 100110, 011001\}$$

وكانت $w = 101010$.

(**١,٦,٩**) إذا كانت $w = 011001$ هي الكلمة المستقبلة فما هي كلمة الشفرة التي تكون على الأرجح أُرسلت من بين الكلمات التالية:

101000 ، 100111 ، 000111 ، 110101 ، 110110 ؟

(**١,٦,١٠**) افترضنا في المبرهنة (**١,٦,٣**) أن $\frac{1}{2} < p < 1$. ماذا يتغير في نص المبرهنة (**١,٦,٣**) لو استبدلنا الفرض ليكون:

(أ) $0 < p < \frac{1}{2}$ (ب) $p = \frac{1}{2}$.

(١,٧) بعض أساسيات الجبر

Some Basic Algebra

أحد أهدافنا هو إيجاد طريقة فعّالة لمعرفة أقرب كلمة شفرة للكلمة المستقبلة. فإذا احتوت الشفرة على عدد كبير من الكلمات فتكون طريقة مقارنة جميع كلمات الشفرة كلمة كلمة مع الكلمة المستقبلة w هي طريقة غير مجدية. على سبيل المثال، إذا كان عدد كلمات الشفرة يساوي 2^{12} (كما هو الحال في رحلات بعض المركبات الفضائية) فيكون من الصعب جداً لطريقة فك التشفير هذه (أي مقارنة الكلمات كلمة كلمة) مواكبة عملية الإرسال. وللتغلب على هذه المشكلة نعرّف بعض العمليات على الشفرات لجعلها نظاماً رياضياً.

لنفرض أن $K = \{0,1\}$ وأن K^n مجموعة جميع الكلمات الثنائية ذات الطول n.

نعرّف الجمع والضرب على K كالتالي:

$$0 + 0 = 1 + 1 = 0$$
$$0 + 1 = 1 + 0 = 1$$
$$0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

ونعرّف الجمع على K^n بجمع الإحداثيات المتقابلة المعرّف على K. فمثلاً، إذا كانت $v = 01101$ و $w = 11001$ فإن:

$$v + w = 01101 + 11001 = 10100$$

من الواضح أن حاصل جمع كلمتين ثنائيتين من الطول n هو كلمة ثنائية من الطول n وبهذا نرى أن K^n مغلق تحت عملية الجمع.

تُسمى عناصر K **أعداداً قياسية (Scalars)** كما هو متفق عليه في الجبر الخطي. وبهذا نعرّف الضرب بعدد قياسي على K^n بضرب كل من إحداثيات الكلمة بالعدد القياسي وحيث إن العددين القياسين هنا هما فقط 0 و 1 فنحصل على ضربين قياسين للكلمة w هما $0.w$ و $1.w$. عناصر الكلمة $0.w$ جميعها أصفار وتُسمى **الكلمة الصفرية (Zero Word)**، وأما الكلمة $1.w$ فتساوي الكلمة w نفسها. ومن الواضح أن K^n مغلق تحت عملية الضرب بعدد قياسي.

باستخدام عمليتي الجمع والضرب بعدد قياسي نستطيع وبسهولة إثبات أن K^n **فضاء متجهات (Vector Space)** أو **فضاء خطي (Linear Space)**. أي أنه يحقق الخواص التالية لكل $u, v, w \in K^n$ ولكل $a, b \in K$:

(١) $v + w \in K^n$.

(٢) $(u + v) + w = u + (v + w)$.

(٣) $v + 0 = 0 + v = v$.

(٤) يوجد $v' \in K^n$ حيث $v + v' = v' + v = 0$.

(٥) $v + w = w + v$.

(٦) $av \in K^n$.

(٧) $a(v + w) = av + aw$.

(٨) $(a + b)v = av + bv$.

(٩) $(ab)v = a(bv)$.

(١٠) $1v = v$.

تمارين

(**١,٧,١**) أثبت أن $v + v = 0$ لكل $v \in K^n$.

(**١,٧,٢**) إذا كان $v, w \in K^n$ وكان $v + w = 0$ فأثبت أن $v = w$.

(**١,٧,٣**) إذا كان $u + v = w$ حيث $u, v, w \in K^n$ فأثبت أن $u + w = v$.

لنفرض أن v أُرسلت عبر قناة BSC وأن w هي الكلمة المستقبلة. إذا كان 0 هو أحد إحداثيات $v + w$ فيكون الإحداثي المقابل في الكلمة v قد أُرسل بدون خطأ. أما إذا كان 1 هو أحد إحداثيات $v + w$ فيكون قد حصل خطأ في إرسال الإحداثي المقابل في الكلمة v. تُسمى الكلمة $v + w$ **نمط الخطأ أو الخطأ (Error Pattern or Error)**. على سبيل المثال، إذا كانت الكلمة المرسلة هي $v = 10101$ وكانت $w = 01100$ فإن نمط الخطأ هو $v + w = 11001$ ونرى وقوع خطأ في إرسال الإحداثيات الأول والثاني والخامس.

(١,٨) الوزن والمسافة
Weight and Distance

نقدم في هذا البند مفهومين مهمين. لنفرض أن v كلمة من الطول n. يعرف **الوزن أو وزن هامينغ (Weight or Hamming Weight)** للكلمة v ويُرمز له بالرمز

$wt(v)$ على أنه عدد مرات ظهور الإحداثي 1 في الكلمة v. فمثلاً، $wt(110101) = 4$ و $wt(00000) = 0$.

لنفرض أن v و w كلمتان من الطول n. تُعرف **مسافة هامينغ أو المسافة (Hamming Distance or Distance)** بين الكلمتين v و w ويُرمز لها بالرمز $d(v,w)$ على أنها عدد المواقع المختلفة بين v و w. فمثلاً، $d(01011\,,00111) = 2$ و $d(10110\,,10110) = 0$.

لاحظ أن المسافة بين v و w تساوي وزن نمط الخطأ $u = v + w$.

أي أن:

$$d(v,w) = wt(v+w).$$

فمثلاً، إذا كانت $v = 11010$ و $w = 01101$ فإن:

$$d(v,w) = d(11010\,,01101) = 4$$

وإن $wt(v+w) = wt(11010 + 01101) = wt(10111) = 4$.

لاحظ أيضاً أنه يمكن إعادة صياغة الاحتمال المقدم في البند (**١,٦**) على النحو التالي:

$$\varphi_p(v,w) = p^{n-wt(u)}(1-p)^{wt(u)}$$

حيث $u = v + w$ هو نمط الخطأ.

نسمي $\varphi_p(u,w)$ **احتمال نمط الخطأ (Probability of Error Pattern)** $u = v + w$.

تمارين

(**١,٨,١**) احسب وزن كل من الكلمات التالية ثم احسب المسافة بين كل زوج منهما:

$v_1 = 1001010$ ، $v_2 = 0110101$ ، $v_3 = 0011110$ ، $v_4 = v_2 + v_3$.

(**١,٨,٢**) لنفرض أن $u = 01011$، $v = 11010$، $w = 01100$. قارن بين أزواج الكلمات فيما يلي:

(أ) $wt(v+w)$ و $wt(v) + wt(w)$

(ب) $d(v,w)$ و $d(v,u) + d(u,w)$

فيما يلي نسرد بعض خواص المسافة والوزن حيث $u,v,w \in K^n$ و $a \in K$:

(١) $0 \le wt(v) \le n$.

(٢) $wt(v) = 0$ إذا وفقط إذا كان $v = 0$.

(٣) $0 \le d(v,w) \le n$.

(٤) $d(v,w) = 0$ إذا وفقط إذا كان $v = w$.

(٥) $d(v,w) = d(w,v)$.

(٦) $wt(v+w) \le wt(v) + wt(w)$.

(٧) $d(v,w) \le d(v,u) + d(u,w)$.

(٨) $wt(av) = a \cdot wt(v)$.

(٩) $d(av,aw) = a \cdot d(v,w)$.

إثبات معظم هذه الخواص واضح من تعريف الوزن والمسافة. في التمرين (**٢,٨,١**) طلبنا من القارئ تقديم أمثلة على الحقيقتين (٦) و (٧). ولبرهان هذه الحقائق حاول استخدام العلاقة $d(v,w) = wt(v+w)$ والتمارين (**١,٧,١**)، (**١,٧,٢**)، (**١,٧,٣**) كلما دعت الحاجة إلى ذلك.

تمارين

(**١,٨,٣**) استخدم K^5 لإيجاد مثال لكل من الخواص التسع المقدمة في بداية الصفحة.

(**١,٨,٤**) أثبت جميع الخواص التسع المقدمة في بداية الصفحة.

سنستخدم هذه الخواص عند الحاجة إليها في البنود القادمة دون ذكر المرجع.

(**١,٩**) فك التشفير الاحتمالي الأقصى
Maximum Likelihood Decoding

نحن الآن جاهزون لمناقشة جادة لمسألتين أساسيتين في نظرية التشفير. سنفترض أننا الطرف المستقبل لقناة BSC حيث نقوم باستقبال رسالة من المرسل الموجود على

الطرف الآخر للقناة. وسنفترض أيضاً أننا الجهة التي قامت بتصميم المرسل. في الحقيقة إن مسألة تصميم المرسل هي من المسائل الأساسية.

هناك كميتان خارج سيطرتنا. أولاهما هي الاحتمال p وهو احتمال إرسال إحداثي عبر قناة BSC دون وقوع خطأ وأما الكمية الثانية فهي عدد الرسائل الممكن إرسالها. في الحقيقة، الرسائل الأصلية ليست بأهم من عدد الرسائل الممكن إرسالها. على سبيل المثال، احتاج **بول ريفير (Paul Revere)** إلى رسالتين فقط قبل قيامه برحلة منتصف الليل المشهورة (سافر بول ريفير بتاريخ ١٨/٤/١٧٧٥م من بوسطن إلى لكسنغتون لتحذير كل من صاموئيل آدمز وجون هانكوك من نية القوات البريطانية لمحاولة اعتقالهم ووصل في الوقت المناسب)[١].

تذكر أن $|S|$ يرمز لعدد عناصر S. ولذا فإن $|K^n| = 2^n$ كما هو مبين في التمرين (**٢,٢,١**). المسألتان الأساسيتان في التشفير هما:

(١,٩,١) التشفير (Encoding)

المطلوب هنا هو تحديد شفرة لغرض استخدامها في إرسال رسائلنا. ولهذا نختار عدداً صحيحاً موجباً k ليكون طول الكلمة الثنائية المقابلة للرسالة المزمع إرسالها. وبما أن الرسائل المختلفة تقابل كلمات ثنائية طول كل منها k فيجب أن نختار k ليحقق $|M| \leq |K^k| = 2^k$.

بعد الانتهاء من اختيار k نقوم بتحديد عدد الإحداثيات المراد إضافتها لكل كلمة من الطول k لضمان تصويب واكتشاف جميع الأخطاء التي نأمل من شفرتنا اكتشافها وتصويبها وهذا هو اختيار كلمات الشفرة وطول الشفرة وليكن n. الآن، لإرسال رسالة معينة يقوم المرسل بإيجاد الكلمة الثنائية من الطول k المقابلة للرسالة

---

(١) المترجمان

ومن ثم إضافة $n-k$ إحداثياً إليها وإرسال كلمة الشفرة ذات الطول n المقابلة للكلمة ذات الطول k.

(١,٩,٢) فك التشفير (Decoding)

نفرض أننا استقبلنا كلمة $w \in K^n$. نُقدم الآن وصفاً لطريقة تُدعى **فك التشفير الاحتمالي الأقصى (Maximum Likelihood Decoding) أو اختصاراً MLD** لتحديد أي من كلمات الشفرة $v \in C$ قد تم إرسالها. يوجد نوعان من MLD هما:

(١) فك التشفير الاحتمالي الأقصى التام (Complete Maximum Likelihood Decoding) أو اختصاراً CMLD يتم فك التشفير في هذا النوع على النحو التالي: إذا وجدت كلمة شفرة وحيدة $v \in C$ هي الأقرب إلى الكلمة المستقبلة w من جميع كلمات الشفرة الأخرى فنعتبر v هي فك تشفير w. أي أنه إذا كان $d(v,w) < d(v_1,w)$ لكل $v_1 \in C$ حيث $v_1 \neq v$ فإن v هي فك تشفير w (أي نعتبر أن v هي كلمة الشفرة المرسلة). أما إذا وجدت عدة كلمات شفرة $v_1, v_2, \cdots, v_k \in C$ تُحقق $d(v_1,w) = d(v_2,w) = \cdots = d(v_k,w)$ وأن $d(v_i,w) < d(u,w)$ لكل $u \in C$ و $v_i \neq u$ فنأخذ أي كلمة v_i من $v_1, v_2, \cdots, v_k$ على أنها فك تشفير w (أي أن v_i هي الكلمة المرسلة).

(٢) فك التشفير الاحتمالي الأقصى غير التام (Incomplete Maximum Likelihood Decoding) أو اختصاراً IMLD يتم فك التشفير في هذا النوع على النحو التالي: إذا وجدت كلمة شفرة وحيدة $v \in C$ هي الأقرب إلى w من جميع كلمات الشفرة الأخرى فتأخذ v لتكون فك تشفير w. أما إذا وجدت عدة كلمات شفرة جميعها تبعد المسافة نفسها عن w والأقرب إلى w من جميع كلمات الشفرة الأخرى فنقوم بطلب إعادة إرسال. وفي بعض الحالات نطلب إعادة إرسال إذا وجدنا أن الكلمة المرسلة w بعيدة عن جميع كلمات الشفرة.

في أمثلة وتمارين هذا البند وفي معظم هذا الكتاب، نستخدم IMLD لفك التشفير. ونريد أن نؤكد على أن طريقة MLD تفشل في بعض الأحيان خاصة إذا وقعت أخطاء كثيرة أثناء الإرسال عبر قناة BSC.

تكون كلمة الشفرة $v \in C$ الأقرب إلى الكلمة المستقبلة w عندما تكون المسافة $d(v,w)$ صغرى، ونرى استناداً إلى المبرهنة (**١,٦,٣**) أن الاحتمال المقرون $\varphi_p(v,w)$ أعظمي، وبهذا تكون v هي على الأرجح كلمة الشفرة المرسلة وهذا موضح في المثال (**١,٦,٤**). وبما أن $d(v,w) = wt(v+w)$ هو وزن نمط الخطأ $u = v + w$ فمن الممكن إعادة صياغة المبرهنة (**١,٦,٣**) لتكون:

$\varphi_p(v_1,w) \leq \varphi_p(v_2,w)$ إذا وفقط إذا كان $wt(v_1+w) \geq wt(v_2+w)$.

وهذا يعني أن الكلمة المرسلة ذات نمط خطأ وزنه أصغري.

وبهذا نرى أن الإستراتيجية المتبعة في طريقة IMLD هي اختبار أنماط الأخطاء $v+w$ لكل كلمة شفرة v ومن ثم اختيار كلمة الشفرة v التي تؤدي إلى نمط خطأ وزنه أصغري.

مثال (١,٩,٣)

لنفرض أن $|M| = 2$ (أي أن $k = 1$) وأننا اخترنا $n = 3$ و $C = \{000, 111\}$. إذا كانت $v = 000$ هي كلمة الشفرة المرسلة فبيّن متى يكون فك التشفير بطريقة IMLD صحيحاً (أي استنتاج أن $v = 000$ هي فعلاً كلمة الشفرة المرسلة) ومتى يكون استنتاج IMLD أن 111 هي الكلمة المرسلة (أي أن يكون الاستنتاج خاطئاً).

الحل

نقوم بإنشاء الجدول (**١,١**) على النحو التالي:

الجدول (١,١). جدول IMLD للمثال (١,٩,٣).

الكلمات المستقبلة	أنماط الأخطاء		فك التشفير (التصويب)[٢]
w	$000+w$	$111+w$	v
000	000*	111	000
100	100*	011	000
010	010*	101	000
001	001*	110	000
110	110	001*	111
101	101	010*	111
011	011	100*	111
111	111	000*	111

العمود الأول من الجدول (١,١) يُبيّن جميع الكلمات الممكن استقبالها وهذه جميع عناصر K^3. العمودان الثاني والثالث يبينان أنماط الأخطاء w لكل كلمة شفرة $v \in C$. وبما أن طريقة IMLD تختار نمط الخطأ الأصغر وزناً فقمنا بمقارنة أوزان أنماط الأخطاء في العمودين الثاني والثالث ووضعنا علامة * بمحاذة نمط الوزن الأصغر. وأخيراً وضعنا في العمود الأخير الكلمة $v \in C$ التي يكون عمود نمط الخطأ w لها معلماً بالعلامة *. وهذه هي كلمة الشفرة v التي تستنتج طريقة IMLD أنها قد أُرسلت عند استقبال الكلمة w المقابلة لها. وبهذا يكون فك تشفير كل من الكلمات المرسلة 000 ، 100 ، 010 ، 001 بطريقة IMLD هي كلمة الشفرة 000 (وهذا استنتاج صائب) وفك تشفير كل من الكلمات المرسلة 110 ، 101 ، 011 ، 111 هي كلمة الشفرة 111 (وهذا استنتاج خاطئ). ▲

(٢) المترجمان: ننبّه القارئ إلى أن التعبير "فك التشفير" المستخدم في هذا الكتاب يعني تصويب الخطأ ويرجع سبب استخدام المؤلفون للتعبير "فك التشفير" عوضاً عن "تصويب الخطأ" إلى تعود مهندسي الاتصالات على هذا الاستخدام للمصطلح. وسنتبنى التعبير الذي استخدمه المؤلفون لهذا المصطلح.

مثال (١,٩,٤)

في هذا المثال نفرض أن $|M| = 3$ وأن $n = 4$ وأن $C = \{0000, 1010, 0111\}$. نقوم بإنشاء جدول IMLD بطريقة مشابهة للمثال (**١,٩,٣**)، والخلاف الوحيد هنا هو أنه إذا وجد أكثر من نمط خطأ واحد ذي وزن أصغري فإننا لن نضع علامة * في الصف الذي يحتوي على هذه الأنماط ونضع العلامة — (تعني لا شيء) في عمود فك التشفير لهذا الصف. هذا يعني أن طريقة IMLD لفك التشفير تطلب إعادة إرسال عند وجود أكثر من نمط خطأ له الوزن الأصغر.

الجدول (١,٢). جدول IMLD للمثال (١,٩,٤).

الكلمات المستقبلة	أنماط الأخطاء			فك التشفير
w	$0111 + w$	$1010 + w$	$0000 + w$	v
0000	0000*	1010	0111	0000
1000	1000	0010	1111	—
0100	0100*	1110	0011	0000
0010	0010	1000	0101	—
0001	0001*	1011	0110	0000
1100	1100	0110	1011	—
1010	1010	0000*	1101	1010
1001	1001	0011	1110	—
0110	0110	1100	0001*	0111
0101	0101	1111	0010*	0111
0011	0011	1001	0100*	0111
1110	1110	0100*	1001	1010
1101	1101	0111	1010*	0111
1011	1011	0001*	1100	1010
0111	0111	1101	0000*	0111
1111	1111	0101	1000*	0111

▲

تمارين

(**١,٩,٥**) لنفرض أن $|M| = 2$، $n = 3$، $C = \{001, 101\}$. إذا كانت $v = 001$ هي كلمة الشفرة المرسلة فمتى يكون استنتاج طريقة IMLD صائباً بأن v هي المرسلة ومتى يكون استنتاج طريقة IMLD خاطئاً بأن 101 هي الكلمة المرسلة ؟

(**١,٩,٦**) لنفرض أن $|M| = 3$ ، $n = 3$. لكل كلمة مرسلة $w \in K^3$ جد كلمة الشفرة v في الشفرة $C = \{000, 001, 110\}$ التي تستنتج طريقة IMLD أنها قد أُرسلت.

(**١,٩,٧**) أنشئ جدول IMLD لكل من الشفرات التالية:

(أ) $C = \{101, 111, 011\}$

(ب) $C = \{000, 001, 010, 011\}$

(ج) $C = \{0000, 0001, 1110\}$

(د) $C = \{0000, 1001, 0110, 1111\}$

(هـ) $C = \{00000, 11111\}$

(و) $C = \{00000, 11100, 00111, 11011\}$

(ز) $C = \{00000, 11110, 01111, 10001\}$

(ح) $C = \{000000, 101010, 010101, 111111\}$

بيّنا في البند الجزئي (**١,٩,١**) أنه يجب علينا اختيار n و C. ولذا تكون بعض الخيارات أفضل من غيرها. نسرد هنا ثلاثة معايير مهمة لقياس الخيارات الجيدة:

(١) كلما ازداد طول الكلمة ازداد الزمن اللازم للإرسال وفك التشفير. وبهذا لا ينصح باختيار عدد كبير n. أي أن معدل المعلومات يجب أن يكون أقرب إلى 1 كلما أمكن ذلك.

(٢) إذا كان عدد الرسائل المرسلة كبيراً (مثلاً $|C|$ يساوي بضعة آلاف) فإن ذلك يستغرق زمناً كبيراً لتنفيذ طريقة IMLD. ولذا فالاختيار المدروس للشفرة C يؤدي إلى طرق ماهرة وسريعة لتنفيذ IMLD.

(٣) إذا وقعت أخطاء كثيرة أثناء عملية الإرسال فلا يصلح استخدام طريقة MLD في فك التشفير؛ وذلك لأن الكلمة التي تَدّعي MLD أنها قد أُرسلت هي ليست

الكلمة المرسلة بالفعل، ولهذا يجب علينا عند استخدام MLD اختيار الشفرة C بحيث يكون احتمال نجاح MLD كبير جداً (سنناقش هذا الاحتمال في البند التالي).

نستطيع القول، بناء على ما تقدم، إن الهدف الرئيس لنظرية التشفير هو إيجاد شفرات C تناسب المعايير الثلاث السابقة. وسيكون معظم جهدنا مركزاً لتحقيق هذا الهدف.

(١,١٠) موثوقية MLD
Reliability of MLD

لنفرض أنه قد تم اختيار كل من n و C. ولنفرض أن BSC قناة ذات احتمال p ولنفرض أن $\theta_p(C,v)$ هو احتمال استنتاج IMLD صواباً بأن v هي الكلمة التي أُرسلت. سنقدم الآن طريقة لحساب $\theta_p(C,v)$.

لتكن $L(v) = \{w \in K^n : d(w,v) < d(w,u) \ \ \forall u \in C, u \neq v\}$. أي أن $L(v)$ هي مجموعة كلمات K^n الأقرب إلى v منها إلى أي كلمة من الكلمات الأخرى للشفرة C. لاحظ أيضاً أن $L(v)$ هي بالضبط مجموعة الكلمات من K^n التي إذا استقبلت فإن IMLD تستنتج صواباً أن v هي بالفعل الكلمة المرسلة. عندئذ، نجد أن:

$$\theta_p(C,v) = \sum_{w \in L(v)} \varphi_p(v,w)$$

من الممكن إيجاد $L(v)$ من جدول IMLD المبين في البند السابق وذلك كما يلي:

في كل صف من صفوف الجدول الذي يحتوي فك التشفير v في عموده الأخير تنتمي الكلمة $w \in K^n$ الواقعة في العمود الأول لهذا الصف تنتمي إلى $L(v)$. وهذه هي جميع كلمات $L(v)$.

لاحظ أيضاً أن $\theta_p(C,v)$ هي مجموع الاحتمالات المأخوذ على الكلمات $w \in L(v)$ بحيث يكون نمط الخطأ الذي تم وقوعه أثناء الإرسال هو $v + w$.

من الممكن استخدام θ_p لمقارنة شفرتين واختيار الشفرة الأفضل منهما (التي تحقق المعيار (٣) من المعايير المقدمة في البند السابق)، مع ملاحظة أن $\theta_p(C,v)$ تتجاهل إمكانية إعادة الإرسال عند تساوي المسافتين بين الكلمة المرسلة وكلمتي شفرة، وهذا يؤدي في بعض الأحيان إلى الخروج عن المألوف (فمثلاً، $\theta_p(K^n,v) > \theta_p(C,v)$ لكل $v \in K^n$ ولكل $u \in C$ حيث C شفرة اختبار النوعية المكوّنة من K^n)، ومع ذلك فهو تقريب أولي مناسب لقياس الموثوقية. من المؤكد أيضاً أن $\theta_p(C,v)$ هو حد أدنى لاحتمال فك تشفير v بشكل صائب.

مثال (١,١٠,١)

لنفرض أن $p = 0.9$ ، $|M| = 2$ ، $n = 3$ ، $C = \{000\,,111\}$ كما هو مبين في المثال (**١,٩,٣**). احسب احتمال أن تكون طريقة IMLD قد استنتجت صواباً أن v هي بالفعل الكلمة التي تم إرسالها بعد عملية إرسال واحدة إذا كانت:

(أ) $v = 000$ (ب) $v = 111$.

الحل

(أ) باستخدام الجدول (**١,١**) نجد أن $v = 000$ هي فك التشفير في الصفوف الأربعة الأولى ونرى أن $L(000)$ (مجموعة كلمات K^3 الأقرب إلى $v = 000$ منها إلى 111) هي:

$$L(000) = \{000\,,100\,,010\,,001\}$$

وبهذا يكون:

$$\begin{aligned}\theta_p(C,000) &= \varphi_p(000{,}000) + \varphi_p(000{,}100) + \varphi_p(000{,}010) + \varphi_p(000{,}001)\\ &= p^3 + p^2(1-p) + p^2(1-p) + p^2(1-p)\\ &= p^3 + 3p^2(1-p)\\ &= .972 \quad (\text{على فرض أن } p = 0.9)\end{aligned}$$

(ب) إذا كانت $v = 111$ فنجد في هذه الحالة أن :

$$L(111) = \{110\,, 101\,, 011\,, 111\}$$

ونرى أن :

$$\begin{aligned}\theta_p(C, 111) &= \varphi_p(111\,, 110) + \varphi_p(111\,, 101) + \varphi_p(111, 011) + \varphi_p(111\,, 111)\\ &= p^2(1-p) + p^2(1-p) + p^2(1-p) + p^3\\ &= 3p^2(1-p) + p^3\\ &= .972 \quad (\text{على فرض أن } p = 0.9)\end{aligned}$$

▲

تمرين

(**١,١٠,٢**) لنفرض أن $p = 0.9$، $|M| = 2$، $n = 3$، $C = \{001\,, 101\}$ كما هو مبيّن في التمرين (**١,٩,٥**).

(أ) إذا كانت $v = 001$ هي الكلمة المرسلة فاحسب احتمال أن تكون طريقة IMLD قد استنتجت صواباً أن v هي بالفعل الكلمة التي تم إرسالها بعد عملية إرسال واحدة.

(ب) أعد الفقرة (أ) للكلمة $v = 101$.

إجابة التمرين (**١,١٠,٢**) في كلتا الحالتين هي $\theta_p(C, v) = 0.900$. وبمقارنة ذلك مع نتيجة المثال (**١,١٠,١**) نجد أن الشفرة $C = \{000\,, 111\}$ أفضل من الشفرة $C = \{001\,, 101\}$ (لأن $0.900 < 0.972$) على الأقل من وجهة نظر المعيار (٣) المقدّم في البند السابق. بناء على ما تقدم نستطيع القول (على الأقل عندما يكون n عدداً صغيراً) إن حساب الاحتمال يُزودنا بمعرفة الحالات التي تكون فيها نتائج طريقة IMLD مرضية. ولحسن الحظ ستكون حسابات الاحتمالات لمعظم الشفرات التي سندرسها لاحقاً أكثر سهولة.

مثال (١,١٠,٣)

لنفرض أن $p = 0.9$، $|M| = 3$، $n = 4$، $C = \{0000, 1010, 0111\}$ كما هو مبين في المثال (**١,٩,٤**). لكل $v \in C$ احسب $\theta_p(C, v)$.

الحل

(أ) في حالة $v = 0000$ لدينا:

$L(0000) = \{0000, 0100, 0001\}$ (من الجدول (**١,٢**))

$$\begin{aligned}\theta_p(C, v) &= \varphi_p(0000, 0000) + \varphi_p(0000, 0100) + \varphi_p(0000, 0001) \\ &= p^4 + p^3(1-p) + p^3(1-p) \\ &= p^4 + 2p^3(1-p) = .8019\end{aligned}$$

(ب) إذا كان $v = 1010$ فإن:

$$L(1010) = \{1010, 1110, 1011\}$$

$$\begin{aligned}\theta_p(C, v) &= \varphi_p(1010, 1010) + \varphi_p(1010, 1110) + \varphi_p(1010, 1011) \\ &= p^4 + p^3(1-p) + p^3(1-p) \\ &= p^4 + 2p^3(1-p) = .8019\end{aligned}$$

(ج) وأخيراً في الحالة $v = 0111$ يكون:

$$L(0111) = \{0110, 0101, 0011, 1101, 0111, 1111\}$$

$$\begin{aligned}\theta_p(C, v) &= \varphi_p(0111, 0110) + \varphi_p(0111, 0101) + \varphi_p(0111, 0011) \\ &\quad + \varphi_p(0111, 1101) + \varphi_p(0111, 0111) + \varphi_p(0111, 1111) \\ &= p^3(1-p) + p^3(1-p) + p^3(1-p) + p^2(1-p)^2 + p^4 + p^3(1-p) \\ &= p^4 + 4p^3(1-p) + p^2(1-p)^2 = .9558\end{aligned}$$

بالتمحيص في هذه الاحتمالات نجد أن احتمال أن تكون طريقة IMLD قد استنتجت صواباً أن الكلمة 0111 هي المرسلة ليس سيئاً. ولكن احتمال أن تكون قد استنتجت صواباً أن الكلمة 0000 أو الكلمة 1010 هي الرسالة فهو سيء للغاية. وبهذا نستنتج (على الأقل من المعيار الثالث المقدم في البند السابق) أن اختيار $C = \{0000, 1010, 0111\}$ ليس بالاختيار المناسب لشفرة. ▲

تمارين

(**١,١٠,٤**) لنفرض أن $p = 0.9$ وأن $C = \{000, 001, 110\}$ كما هو مبين في التمرين (**١,٩,٦**). إذا كانت $v = 110$ هي الكلمة المرسلة فاحسب احتمال أن تكون طريقة IMLD استنتجت صواباً أن v هي بالفعل الكلمة المرسلة واحسب احتمال أن تكون طريقة IMLD استنتجت خطأ أن الكلمة 000 هي المرسلة.

(**١,١٠,٥**) لكل من الشفرات C أدناه، احسب $\theta_p(C, v)$ لكل $v \in C$ مستخدماً $p = 0.9$ (جداول IMLD لهذه الشفرات قد تم إنشاؤها في التمرين (**١,٩,٧**)):

(أ) $C = \{101, 111, 011\}$

(ب) $C = \{000, 001, 010, 011\}$

(ج) $C = \{0000, 0001, 1110\}$

(د) $C = \{0000, 1001, 0110, 1111\}$

(هـ) $C = \{00000, 11111\}$

(و) $C = \{00000, 11100, 00111, 11011\}$

(ز) $C = \{00000, 11110, 01111, 10001\}$

(ح) $C = \{000000, 101010, 010101, 111111\}$.

(١,١١) شفرات اكتشاف الأخطاء

Error-Detecting Codes

في هذا البند نعرّف بشكل دقيق متى يكون بمقدور شفرة C اكتشاف الأخطاء. تذكر أنه إذا كانت $v \in C$ هي الكلمة المرسلة وكانت $w \in K^n$ هي الكلمة المستقبلة فإن $u = v + w$ هو نمط الخطأ. لاحظ أيضاً أن أي كلمة $u \in K^n$ ممكن أن تكون نمط خطأ والمطلوب هو معرفة أي من أنماط الأخطاء تستطيع الشفرة C اكتشافها.

نقول إن الشفرة C **تكتشف (Detects)** نمط الخطأ u إذا وفقط إذا كان $v+u \notin C$ لكل $v \in C$. أي أن C تستطيع اكتشاف نمط الخطأ u إذا تحقق ما يلي:

بعد إرسال كلمة الشفرة v واستقبال $v+u$ يكون بإمكان **فاكك التشفير (Decoder)** التحقق من أن $v+u$ ليست كلمة شفرة ومن ثم فهناك خطأ ما قد وقع.

مثال (١,١١,١)

لنفرض أن $C=\{001, 101, 110\}$. بيّن أن C تكتشف نمط الخطأ $u=010$ ولكنها لا تكتشف نمط الخطأ $u=100$.

الحل

إذا كان $u=010$ فنقوم بحساب $v+010$ لكل $v \in C$ فنجد:

$$001+010=011 \notin C$$
$$101+010=111 \notin C$$
$$110+010=100 \notin C$$

وبهذا نرى أن C تكتشف $u=010$. وبحساب $v+100$ لكل $v \in C$ نجد:

$$001+100=101 \in C$$
$$101+100=101 \in C$$
$$110+100=010 \notin C$$

وبما أن واحداً على الأقل من هذه المجاميع ينتمي إلى C فنرى أن C لا تكتشف نمط الخطأ $u=100$. ▲

تمارين

(١,١١,٢) لنفرض أن $C=\{001, 101, 110\}$. بيّن ما إذا كان بإمكان C إكتشاف أي من أنماط الأخطاء التالية:

(أ) 011 (ب) 001 (ج) 000

(١,١١,٣) لكل من الشفرات التالية C بيّن فيما إذا كان بإمكان C اكتشاف نمط الخطأ u المعطى:

(أ) $C = \{00000, 10101, 00111, 11100\}$

(i) $u = 10101$
(ii) $u = 01010$
(iii) $u = 1010$

(ب) $C = \{1101, 0110, 1100\}$

(i) $u = 0010$
(ii) $u = 0011$
(iii) $u = 1010$

(ج) $C = \{1000, 0100, 0010, 0001\}$

(i) $u = 1001$
(ii) $u = 1110$
(iii) $u = 0110$

(١,١١,٤) أي من أنماط الأخطاء تستطيع الشفرة $C = K^n$ اكتشافها ؟

(١,١١,٥) (أ) لتكن C شفرة تحتوي الكلمة الصفرية. أثبت أنه إذا كان نمط الخطأ u هو كلمة شفرة فليس بإمكان C اكتشاف u.

(ب) أثبت عدم وجود شفرات يكون بإمكانها اكتشاف نمط الخطأ $u = 0$.

من الممكن استخدام جدول IMLD لمعرفة أنماط الأخطاء التي تستطيع الشفرة C اكتشافها. فالعمود الأول يسرد جميع كلمات K^n ومن ثم يمكن النظر إليه على أنه جميع أنماط الأخطاء الممكنة، وبذلك تكون أعمدة أنماط الأخطاء في الجدول هي المجموع $v + u$ لكل $v \in C$. الآن، إذا لم ينتم أي من هذه المجاميع في صف ما إلى الشفرة C فإن C تكتشف نمط الخطأ الواقع في العمود الأول من ذلك الصف.

مثال (١,١١,٦)

لنفرض أن $C = \{000, 111\}$ هي الشفرة حيث جدول IMLD لها هو الجدول **(١,١)**. العمود الأول يحتوي على جميع أنماط الأخطاء u الممكنة. ولكل u نجد أن

جميع المجاميع $v+u$ لكل $v \in C$ هي الموجودة في العمودين الثاني والثالث. فإذا كانت جميع هذه المجاميع لا تنتمي إلى C (أي لا تساوي 000 أو 111) فإن C يكتشف u. وبهذا نرى أن C تكتشف أنماط الأخطاء 100، 010، 001، 110، 101، 011 وذلك بالنظر إلى الأعمدة من 2 إلى 7، ولكن لا تستطيع C إكتشاف أي من الخطأين 000 أو 111. ▲

تمرين

(١,١١,٧) استخدم جدول IMLD للشفرة C المنشأ في التمرين **(١,٩,٧)** لإيجاد أنماط الأخطاء التي تستطيع C اكتشافها.

طريقة أخرى ولكنها أسرع كثيراً لإيجاد أنماط الأخطاء التي يمكن لشفرة C اكتشافها تكون بإيجاد أنماط الأخطاء التي لا تستطيع C اكتشافها ومن ثم فإن C تكتشف جميع أنماط الأخطاء المتبقية. من الواضح أنه إذا كانت $v,w \in C$ وكان $e = v + w$ فلا تستطيع C اكتشاف e؛ وذلك لأن $v + e = w \in C$. وبهذا نرى أن أنماط الأخطاء التي لا تتمكن C من اكتشافها هي مجموعة جميع الكلمات التي يمكن كتابتها كمجموع كلمتي شفرة.

مثال (١,١١,٨)

لنفرض أن $C = \{000, 111\}$ شفرة. بملاحظة أن

$$000 + 000 = 000$$
$$000 + 111 = 111$$
$$111 + 111 = 000$$

نجد أن مجموعة أنماط الأخطاء التي لا تكتشفها C هي $\{000, 111\}$. وبهذا فإن مجموعة أنماط الأخطاء التي تستطيع C اكتشافها هي $K^3 \backslash \{000, 111\}$. ▲

مثال (١,١١,٩)

لنفرض أن $C = \{1000, 0100, 1111\}$ شفرة. بما أن:

$$1000 + 1000 = 0000$$
$$1000 + 0100 = 1100$$
$$1000 + 1111 = 0111$$
$$0100 + 1111 = 1011$$

فنجد أن مجموعة أنماط الأخطاء التي لا تستطيع C اكتشافها هي $\{0000, 1100, 0111, 1011\}$ وبهذا تكون مجموعة أنماط الأخطاء التي يتم اكتشافها بواسطة C هي $K^4 \setminus \{0000, 1100, 0111, 1011\}$. ▲

تمرين

(١,١١,١٠) جد مجموعة أنماط الأخطاء التي يمكن اكتشافها بواسطة كل من الشفرات التالية ثم قارن إجاباتك مع إجابات التمرين **(١,١١,٧)**:

(أ) $C = \{101, 111, 011\}$

(ب) $C = \{000, 001, 010, 011\}$

(ج) $C = \{0000, 0001, 1110\}$

(د) $C = \{0000, 1001, 0110, 1111\}$

(هـ) $C = \{00000, 11111\}$

(و) $C = \{00000, 11100, 00111, 11011\}$

(ز) $C = \{00000, 11110, 01111, 10001\}$

(ح) $C = \{000000, 101010, 010101, 111111\}$.

سنبين الآن طريقة أخرى لتحديد بعض أنماط الأخطاء التي تستطيع شفرة C اكتشافها بدون الحاجة إلى إجراء الحسابات الطويلة ولكننا نقدم أولاً عدداً آخر يقترن بالشفرة C.

إذا كانت C شفرة تحتوي كلمتين على الأقل فتعرّف مسافة C **(Distance of C)** على أنها أصغر مسافة $d(v,w)$ لكل $v,w \in C$ حيث $v \neq w$. وبما أن $d(v,w) = wt(v+w)$ فتكون مسافة الشفرة C أصغر الأوزان $wt(v+w)$ لكل $v,w \in C$ حيث $v \neq w$.

تشترك مسافة الشفرة مع المسافة الاقليدية بالعديد من الخواص وهذا يساعد على فهم مفهوم مسافة الشفرة.

مثال (١,١١,١١)

إذا كانت $C = \{0000\,,1010\,,0111\}$ فإن:

$$d(0000\,,1010) = 2$$
$$d(0000\,,0111) = 3$$
$$d(1010\,,0111) = 3$$

وبهذا تكون مسافة الشفرة C تساوي 2. ▲

تمارين

(١,١١,١٢) احسب مسافة كل من الشفرات التالية:

(أ) $C = \{101\,,111\,,011\}$

(ب) $C = \{000\,,001\,,010\,,011\}$

(ج) $C = \{0000\,,0001\,,1110\}$

(د) $C = \{0000\,,1001\,,0110\,,1111\}$

(هـ) $C = \{00000\,,11111\}$

(و) $C = \{00000\,,11100\,,00111\,,11011\}$

(ز) $C = \{00000\,,11110\,,01111\,,10001\}$

(ح) $C = \{000000\,,101010\,,010101\,,111111\}$

(١,١١,١٣) احسب مسافة الشفرة التي نحصل عليها بإضافة إحداثي اختبار النوعية للمجموعة K^n.

المبرهنة التالية تساعدنا على معرفة الكثير من أنماط الأخطاء التي يمكن لشفرة اكتشافها.

مبرهنة (١,١١,١٤)

تستطيع شفرة C مسافتها d اكتشاف على الأقل جميع أنماط الأخطاء غير الصفرية التي وزنها لا يزيد عن $d-1$. إضافة إلى ذلك يوجد على الأقل نمط خطأ واحد وزنه d لا تتمكن الشفرة C من اكتشافه.

البرهان

لنفرض أن u نمط خطأ غير صفري حيث $wt(u) \leq d-1$. ولنفرض أن $v \in C$. عندئذ:

$$d(v, v+u) = wt(v+v+u) = wt(u) < d.$$

وبما أن مسافة C تساوي d فإن $v+u \notin C$ وبهذا تستطيع C اكتشاف u. من تعريف المسافة d، نرى وجود كلمتين $v, w \in C$ حيث $d(v,w) = d$. لنفرض أن $u = v + w$ نمط خطأ. عندئذ، $w = v + u \in C$ ومن ثم C لا تكتشف نمط الخطأ u ذا الوزن d. ■

ملحوظة

لاحظ إمكانية اكتشاف الشفرة C لأنماط أخطاء وزنها أكبر من أو يساوي d ولكنها لا تستطيع اكتشاف جميع أنماط الأخطاء ذات الوزن d.

نقول إن شفرة C من الدرجة t في اكتشاف الأخطاء **(t Error-Detecting Code)** إذا تمكنت من اكتشاف جميع أنماط الأخطاء ذات الأوزان التي لا تزيد عن t ولكنها لا تكتشف على الأقل نمط خطأ واحد وزنه $t+1$. وبهذا نرى استناداً إلى المبرهنة **(١,١١,١٤)** أن الشفرة C ذات المسافة d هي شفرة تكتشف أنماط أخطاء من النوع $d-1$.

مثال (١,١١,١٥)

مسافة الشفرة $C = \{000, 111\}$ تساوي $d = 3$. واستناداً إلى المبرهنة (**١,١١,١٤**) نرى أن C تكتشف جميع أنماط الأخطاء ذات الوزن 1 أو الوزن 2 وأن C لا تكتشف نمط الخطأ الوحيد 111 ذا الوزن 3. نمط الخطأ الوحيد الذي لا نستطيع تطبيق المبرهنة (**١,١١,١٤**) عليه هو 000 ولكننا رأينا في التمرين (**١,١١,١٥**) أن نمط الخطأ هذا لا يمكن اكتشافه. ▲

كما أسلفنا فالمبرهنة (**١,١١,١٤**) لا تمنع شفرة C من اكتشاف أنماط أخطاء وزنها d أو أكثر. في العادة، C تكتشف بعض أنماط أخطاء ذات أوزان أكبر من أو تساوي d.

مثال (١,١١,١٦)

مسافة الشفرة $C = \{001, 101, 110\}$ هي $d = 1$. بما أن $d - 1 = 0$ فلا نستطيع استخدام المبرهنة (**١,١١,١٤**) لاكتشاف أنماط الأخطاء ولكنها تضمن لنا وجود نمط خطأ واحد على الأقل وزنه $d = 1$ لا تستطيع C اكتشافه. وكما رأينا في المثال (**١,١١,١**) فنمط الخطأ 100 مثال لذلك. لاحظ أيضاً أن C لا تكتشف نمط الخطأ 010 حيث وزنه 1 أيضاً. ▲

تمارين

(**١,١١,١٧**) مسافة الشفرة $C = \{0000, 1010, 0111\}$ هي $d = 2$. استخدم التمرين (**١,١١,٥**) لإثبات أن C لا تكتشف نمط الخطأ 1010. أثبت أيضاً أن نمط الخطأ هذا هو الوحيد ذو الوزن 2 الذي لا يمكن اكتشافه بواسطة C. جد جميع أنماط الأخطاء التي تستطيع C إكتشافها.

(**١,١١,١٨**) جد جميع أنماط الأخطاء التي تستطيع الشفرة C_3 المقدمة في المثال (**١,٣,٣**) اكتشافها. لاحظ أن C_3 تكتشف أنماط أخطاء من النوع 1.

(**١,١١,١٩**) لكل شفرة C من شفرات التمرين (**١,١١,١٢**) جد جميع أنماط الأخطاء المضمون اكتشافها من قبل C بتطبيق المبرهنة (**١,١١,١٤**).

(**١,١١,٢٠**) لنفرض أن C هي الشفرة التي تحتوي جميع الكلمات ذات الطول 4 وذات الوزن الزوجي. عيّن جميع أنماط الأخطاء التي يمكن اكتشافها من قبل C.

(١,١٢) شفرات تصويب الأخطاء
Error-Correcting Codes

لنفرض أنه قد تم إرسال كلمة الشفرة $v \in C$ عبر قناة BSC ولنفرض أن w هي الكلمة المستقبلة ولنفرض أنه قد وقع نمط الخطأ $u = v + w$ أثناء عملية الإرسال. عندئذ، يكون استنتاج طريقة IMLD صائباً بأن v هي بالفعل الكلمة المرسلة إذا كانت w أقرب إلى v من أي كلمة شفرة أخرى. وإذا حصل هذا في كل مرة يقع فيها نمط الخطأ u بغض النظر عن الكلمة المرسلة فنقول إن C **تصوّب نمط الخطأ** u **(*C* Corrects the Error-Pattern *u*)**. أي أن C تصوّب نمط الخطأ u إذا كان لكل $v \in C$، $v + u$ أقرب إلى v من أي من كلمات الشفرة C الأخرى. ومن الممكن صياغة ذلك رياضياً بقولنا إن C تصوّب نمط الخطأ u إذا تحقق ما يلي :

$$\forall\ v \in C\ ,\forall\ w \in C(v \neq w \rightarrow d(v, v+u) < d(w, v+u)$$

ونقول إن الشفرة C **من الدرجة** t **في تصويب الأخطاء (*t* Error-Correcting Code)** إذا كانت C تصوّب جميع أنماط الأخطاء ذات الوزن t ولا تصوّب نمط خطأ واحد على الأقل وزنه $t + 1$.

مثال (١,١٢,١)

لنفرض أن $C = \{000\ , 111\}$.

(أ) أثبت أن C تصوّب نمط الخطأ $u = 010$.

(ب) أثبت أن C لا تصوّب نمط الخطأ $u = 110$.

الحل

(أ) عند $v = 000$ لدينا :

$$d(000, v+u) = d(000,010) = 1$$
$$d(111, v+u) = d(111,010) = 2$$

وعند $v = 111$ لدينا :

$$d(000, v+u) = d(000,101) = 2$$
$$d(111, v+u) = d(111,101) = 1$$

إذن، C تصوّب نمط الخطأ $u = 010$.

(ب) عند $v = 000$ لدينا :

$$d(000, v+u) = d(000,110) = 2$$
$$d(111, v+u) = d(111,110) = 1$$

وبما أن $v+u$ ليس أقرب إلى $v = 000$ منه إلى $v = 111$ فنرى أن C لا تصوّب نمط الخطأ $u = 110$. ▲

من الممكن الاستعانة بجدول IMLD لمعرفة أنماط الأخطاء التي يمكن تصويبها بواسطة الشفرة C. بالنظر إلى أي عمود نمط خطأ نجد أن كل نمط من أنماط الأخطاء الممكنة (أي كلمات K^n) يقع مرة واحدة في هذا العمود (إذا وقع نمط الخطأ u مرتين في العمود لكلمة شفرة واحد v فإن u تقع في صفين يقابلان كلمتين مستقبلتين مختلفتين w_1 و w_2. وبهذا يكون $u = v + w_1 = v + w_2$ وهذا مستحيل ؛ لأن $w_1 \neq w_2$). كما أن العلامة * في جدول IMLD توضع بجانب نمط الخطأ u في عمود يقابل كلمة الشفرة v إذا كانت $v+u$ أقرب إلى v منها إلى أي كلمة شفرة أخرى. وبهذا نرى أن C تصوّب نمط الخطأ u إذا وجدت العلامة * بجانب u في جميع أعمدة أنماط الأخطاء في جدول IMLD.

مثال (١,١٢,٢)

جدول IMLD للشفرة $C = \{000\ , 111\}$ مبين في الجدول (١,١). توجد علامة * بجانب نمط الخطأ 010 في جميع صفوف وقوعه (الصفان 3 و 6) ولذا فإن طريقة IMLD

تستنتج صواباً أن الكلمة v هي المرسلة ومن ثم فإن C تصوّب نمط الخطأ 010. أما بالنسبة لنمط الخطأ 110 فلا توجد علامة * بجانبه في صف واحد على الأقل من صفوف وقوعه (الصف 4). ونرى أنه لو كانت 111 هي الكلمة المرسلة وأن 001 هي الكلمة المستقبلة فإن طريقة IMLD تستنتج خطأ أن الكلمة المرسلة هي 000. ولذا فإن C لا تصوّب نمط الخطأ 110. لاحظ أيضاً أن C تصوّب جميع أنماط الأخطاء 000، 100، 010، 001 الموضوع بجانبها علامة * في جميع صفوف وقوعها. وبهذا تكون C شفرة تصوّب أنماط الأخطاء من النوع 1. ▲

مثال (١,١٢,٣)

جدول IMLD للشفرة $C = \{0000, 1010, 0111\}$ مبين في الجدول (**١,٢**). يقع نمط الخطأ $u = 1010$ في الصفوف 1، 7، 13 والوقوع الوحيد الموضوع بجانبه علامة * هو الوقوع في الصف 13. ولذا فإن C لا تصوّب نمط الخطأ $u = 1010$. ولكن C تصوّب جميع أنماط الأخطاء 0000، 0100، 0001. ▲

مثال (١,١٢,٤)

هل الشفرة $C = \{001, 101, 110\}$ تصوّب نمط الخطأ $u = 100$ ؟

الحل

صفوف جدول IMLD التي يظهر فيها نمط الخطأ $u = 100$ مبينة في الجدول التالي.

الكلمات المستقبلة	أنماط الأخطاء			فك التشفير
w	$101 + w$	$001 + w$	$110 + w$	v
101	100	000*	011	101
001	000*	100	111	001
010	011	111	100*	110

بالنظر إلى الجدول نجد أن الوقوع الوحيد لنمط الخطأ $u = 100$ الموضوع بجانبه * هو الوقوع في الصف الثالث. ولذا فإن C لا تصوّب نمط الخطأ $u = 100$. ▲

تمارين

(**١,١٢,٥**) لتكن $C = \{001\ ,101\ ,110\}$. هل تصوّب الشفرة C نمط الخطأ $u = 100$ ؟ ماذا عن نمط الخطأ $u = 000$ ؟

(**١,١٢,٦**) أثبت أن نمط الخطأ نفسه لا يمكن أن يقع في صفين مختلفين من جدول IMLD.

(**١,١٢,٧**) أثبت أن جميع الشفرات تصوّب نمط الخطأ صفر.

(**١,١٢,٨**) أي من أنماط الأخطاء تصوبها الشفرة $C = K^n$ ؟

من الممكن استخدام مفهوم مسافة الشفرة لتصميم اختبار لتصويب الأخطاء عوضاً عن استخدام طريقة IMLD الشّاقة أحياناً. وهذا الاختبار هو فحوى المبرهنة التالية. تذكر أن الرمز $\lfloor x \rfloor$ يعني أكبر عدد صحيح لا يزيد عن العدد الحقيقي x. فمثلاً، $\lfloor 5/2 \rfloor = 2$، $\lfloor 3 \rfloor = 3$، $\lfloor 1/2 \rfloor = 0$.

مبرهنة (١,١٢,٩)

لتكن C شفرة مسافتها d. عندئذ، C تصوّب جميع أنماط الأخطاء التي وزنها لا يزيد عن $\lfloor (d-1)/2 \rfloor$. إضافة إلى ذلك يوجد على الأقل نمط خطأ واحد وزنه $1 + \lfloor (d-1)/2 \rfloor$ لا تصوّبه الشفرة C.

البرهان

لنفرض أن u نمط خطأ حيث $wt(u) \leq (d-1)/2$. وليكن $v, w \in C$ حيث $w \neq v$.

سنبرهن أن $d(v, v+u) < d(w, v+u)$. الآن

$$d(w, v+u) + wt(u) = d(w, v+u) + d(v+u, v) \qquad \text{(تعريف الوزن)}$$

$$\geq d(w, v) \qquad \text{(متباينة المثلث)}$$

$$\geq d \qquad \text{(تعريف مسافة الشفرة)}$$

$$\geq 2wt(u) + 1 \qquad \text{(الفرض)}$$

إذن، $d(w, v+u) \geq wt(u)+1 > wt(u) = d(v, v+u)$. وبهذا نرى أن C تصوّب نمط الخطأ u.

نفرض الآن أن $v, w \in C$ حيث $d(v,w) = d$ (لاحظ أن $wt(v+w) = d$).

ولنفرض أن u هو نمط الخطأ الذي نحصل عليه من $v+w$ بتغيير عدد $d-1-\lfloor (d-1)/2 \rfloor$ من الواحدات إلى أصفار. عندئذ

$$wt(u) = d - (d-1-\lfloor (d-1)/2 \rfloor = 1 + \lfloor (d-1)/2 \rfloor$$

سنبرهن الآن أن C لا تصوّب u وذلك بإثبات أن $d(v, v+u) \geq d(w, v+u)$.

ندرس الحالتين: d فردي و d زوجي.

لنفرض أولاً أن d فردي. أي أن $d = 2t+1$ حيث $t \in \mathbb{Z}$. في هذه الحالة يكون $\lfloor (d-1)/2 \rfloor = t$. من ذلك نرى أن:

$$\begin{aligned} d(v, v+u) &= wt(u) = 1+t \\ d(w, v+u) &= wt(w+v+u) = d(v+w, u) \\ &= d - (1 + \lfloor (d-1)/2 \rfloor) \\ &= (2t+1) - (1+t) = t \end{aligned}$$

وبهذا نرى في هذه الحالة أن $d(w, v+u) \geq d(w, v+u)$.

أما إذا كان d زوجياً فإن $d = 2t$ ويكون $\lfloor (d-1)/2 \rfloor = t-1$. من ذلك نجد أن:

$$\begin{aligned} d(v, v+u) &= wt(u) = 1 + (t-1) = t \\ d(w, v+u) &= wt(w+v+u) = d(v+w, u) \\ &= d - (1 + \lfloor (d-1)/2 \rfloor) \\ &= 2t - (1 + (t-1) \\ &= t \end{aligned}$$

وبهذا نستنتج أن $d(v, v+u) \geq d(w, v+u)$. وعليه نرى أن $v+u$ ليس أقرب إلى v منه إلى w ونخلص إلى أن C لا تصوّب نمط الخطأ u. ■

من الواضح، استناداً إلى المبرهنة (**١,١٢,٩**) أن الشفرة ذات المسافة d هي شفرة تصويب أنماط أخطاء من النوع $\lfloor (d-1)/2 \rfloor$.

مثال (١,١٢,١٠)

مسافة الشفرة $C = \{000, 111\}$ هي $d = 3$. بما أن $\lfloor (d-1)/2 \rfloor = 1$ فنجد استناداً إلى المبرهنة (**١,١٢,٩**) أن C تصوّب جميع أنماط الأخطاء من الوزن 0 أو 1. وكما رأينا من المثال (**١,١٢,١**) فإن C تصوّب فعلاً أنماط الأخطاء 000، 100، 010، 001. وزن نمط الخطأ 110 يساوي $\lfloor (d-1)/2 \rfloor + 1 = 2$ وسبق وأن رأينا أن C لا تصوّب نمط الخطأ هذا. ▲

لاحظ أن المبرهنة (**١,١٢,٩**) لا تمنع شفرة C مسافتها d من تصويب أنماط أخطاء وزنها أكبر من $\lfloor (d-1)/2 \rfloor$.

مثال (١,١٢,١١)

مسافة الشفرة $C = \{001, 101\}$ هي $d = 1$ ووزن نمط الخطأ $u = 011$ هو 2 وهذا أكبر من $\lfloor (d-1)/2 \rfloor + 1 = 1$ ومع ذلك فالشفرة C تصوّب نمط الخطأ $u = 011$ كما هو موضّح في جزء جدول IMLD التالي.

w	$001 + w$	$101 + w$	v
010	011^*	111	001
110	111	011^*	101

تمارين

(**١,١٢,١٢**) لكل من الشفرات التالية:

(i) عيّن أنماط الأخطاء التي تصوّبها C. أُنشئت جداول IMLD لهذه الشفرات في التمرين (**١,٩,٧**).

(ii) عيّن أنماط الأخطاء التي تضمن المبرهنة (**١,١٢,٩**) تصويبها بواسطة C.

(أ) $C = \{101, 111, 011\}$

(ب) $C = \{000, 001, 010, 011\}$

(ج) $C = \{0000, 0001, 1110\}$

(د) $C = \{0000, 1001, 0110, 1111\}$

(هـ) $C = \{00000, 11111\}$

(و) $C = \{00000, 11100, 00111, 11011\}$

(ز) $C = \{00000, 11110, 01111, 10001\}$

(ح) $C = \{000000, 101010, 010101, 111111\}$

(١,١٢,١٣) استخدم طريقة المثال **(١,١٢,١١)** لتحديد فيما إذا كانت الشفرة المعطاة تصوّب أنماط الأخطاء المعطاة.

(أ) $C = \{000000, 100101, 010110, 001111, 110011, 101010, 011001, 111100\}$

(i) $u = 001000$

(ii) $u = 000010$

(iii) $u = 100100$

(ب) $C = \{1001011, 0110101, 1110010, 1111111\}$

(i) $u = 0100000$

(ii) $u = 0101000$

(iii) $u = 1100000$

(١,١٢,١٤) لكل شفرة من شفرات التمرين **(١,١٢,١٢)**، جد نمط خطأ من الوزن $1 + \lfloor (d-1)/2 \rfloor$ لا تصوّبه الشفرة.

(١,١٢,١٥) لتكن C شفرة تحتوي جميع الكلمات ذات الطول 4 وذات الوزن الزوجي. عيّن أنماط الأخطاء التي تصوّبها C.

(١,١٢,١٦) لنفرض أن u_1 و u_2 نمطا خطأ طول كل منهما يساوي n ولنفرض أن u_1 و u_2 يتفقان على الأقل في المواقع التي تكون فيها إحداثيات u_1 تساوي 1. إذا كانت C تصوّب u_2 فأثبت أنها تصوّب u_1 أيضاً.

لاحظنا سابقاً أن احتمال وقوع أنماط الأخطاء ذوات الأوزان الصغيرة أكثر من احتمال وقوع أنماط الأخطاء ذوات الأوزان الكبيرة (المبرهنة (**١,٦,٣**)). وبناء على ذلك فعند تصميم الشفرات يجب التركيز على تصميم شفرات يكون بمقدورها تصويب أو على الأقل اكتشاف أنماط الأخطاء ذوات الأوزان الصغيرة.