جامعة المنارة

كلية الهندسة

قسم ميكاترونيكس

Information System Security

أمن نظم المعلومات

مدرسة المقرر

د. بشرى علي معلا

عناوين المحاضرة السابعة

- مقدمة
- خوارزمية (Rivest Cipher 4)RC4:
  - ✓ تعريف بخوارزمية RC4 و استخداماتها
  - ✓ الفكرة الرئيسة لـ RC4
  - ✓ الخطوات التفصيلية لآلية عمل RC4
- تابع البعثرة (HASH FUNCTION):
  - ✓ مقدمة
  - ✓ تعريف تابع البعثرة (Hash function)
  - ✓ خصائص تابع البعثرة
  - ✓ أهم استخدامات تابع البعثرة
  - ✓ تصنيف تابع البعثرة

مقدمة

- تقسم خوارزميات التشفير المتناظر إلى :
  - ✓ خوارزميات التشفير الكتلي: النص الصريح مقسم إلى كتل ذات طول معين (DES)
  - ✓ خوارزميات التشفير التسلسلي :
    - النص الصريح هو سلسلة من البتات/ بايتات (RC4)
    - المفتاح هو سلسلة من البتات /بايتات ناتج عن عملية توليد مطبقة على مفتاح الدخل
    - عملية التشفير تكون باستخدام XOR

➢ مثال:

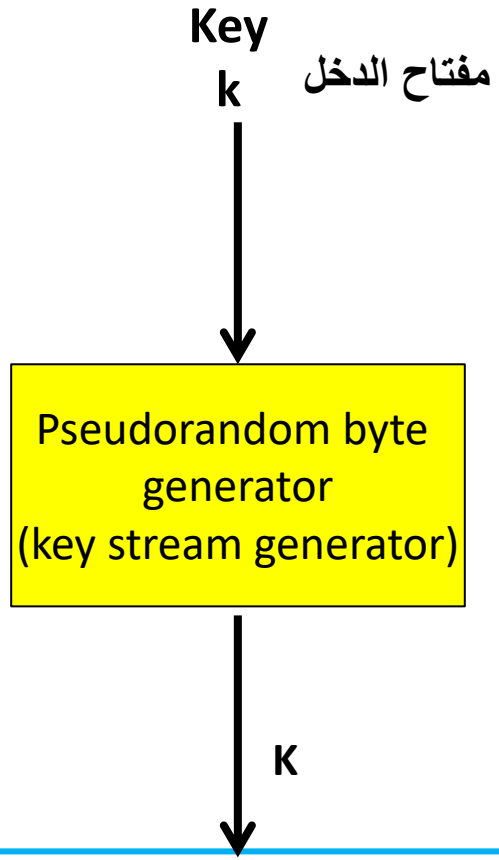
- النص الصريح: 10000101010
- سلسلة المفتاح : 11001010101

$$\begin{array}{r} 10000101010 \\ 11001010101 \oplus \\ \hline 01001111111 \end{array}$$

تعريف بخوارزمية RC4 و استخداماتها

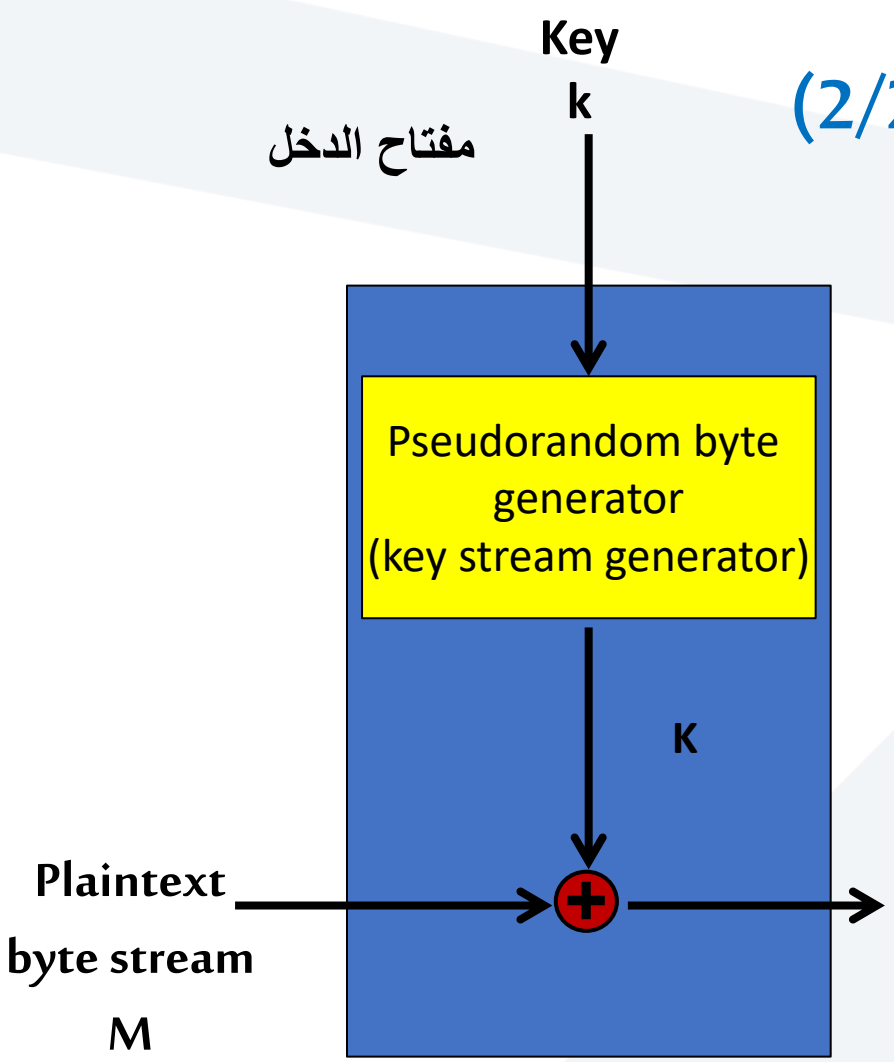
- صممت هذه الخوارزمية في عام 1987 من قبل Ron Rivest
- تعد RC4 نظام التعمية التسلسلي الأكثر انتشاراً
- تستخدم RC4 طول مفتاح متغير، و تطبق العمليات على البايتات بشكل منفرد
- تعتمد هذه الخوارزمية على استخدام التبديل العشوائي للمواقع
- يحتاج هذا النظام من 8 إلى 16 عملية حسابية لإخراج بايت واحد
- **استخداماتها:**
  - ✓ في البروتوكول SSL/TLS المستخدم بين مستكشفات الويب
  - ✓ في المعيار IEEE 802.11 wireless LAN :
  - ✓ ضمن البروتوكولين:
    - ❖ WEP (Wired Equivalent Privacy)
    - ❖ WPA (WiFi Protocol Access)

الفكرة الرئيسة لـ RC4(1/2)

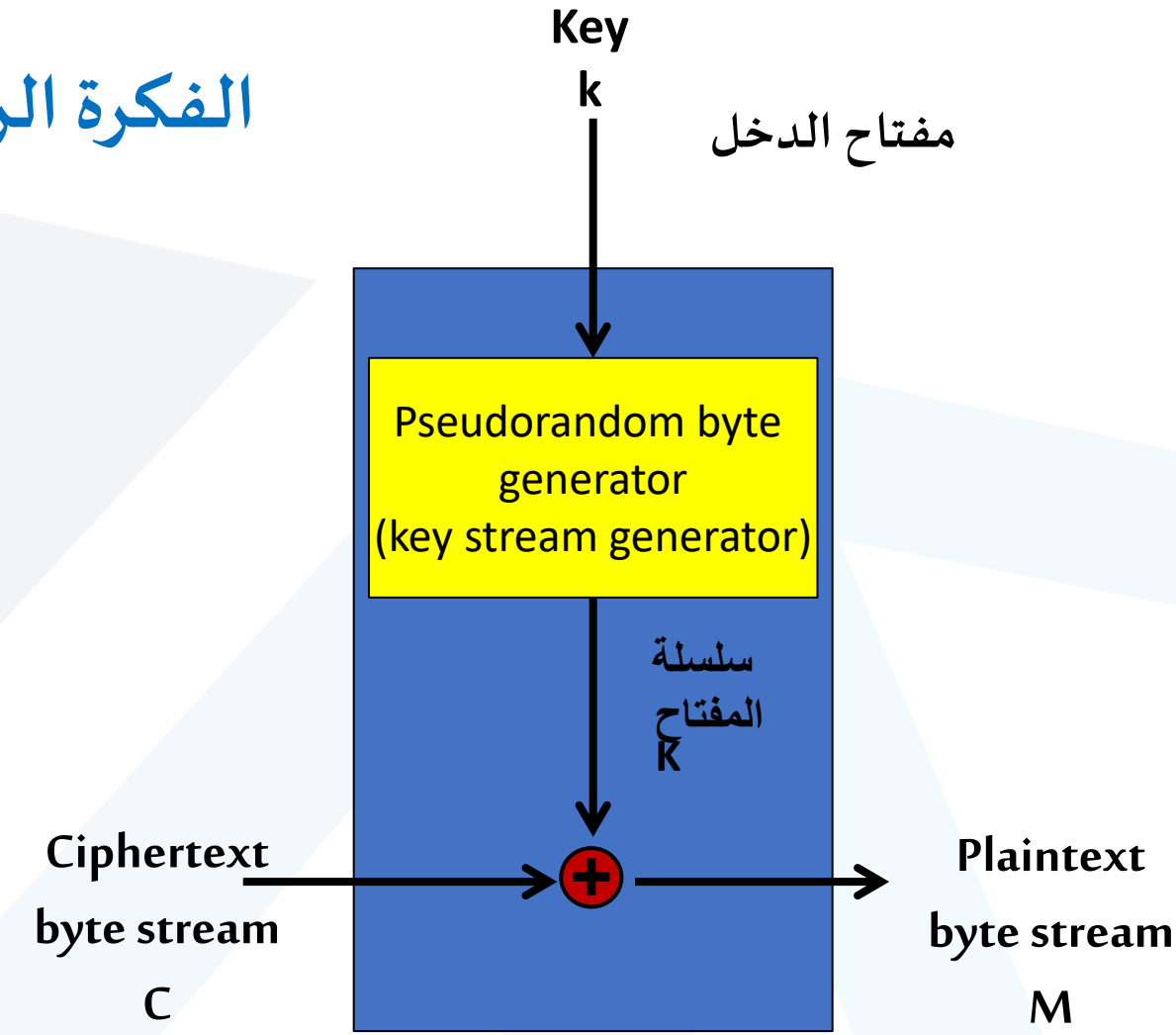


- يعتمد على تعمية النصوص الصريحة على شكل بايت واحد في كل عملية (بنفس الوقت)
- يمثل المفتاح في هذه البنية دخل مولد خانات شبه عشوائي Pseudorandom byte generator
  - ✓ ينتج المولد سلسلة من الأعداد العشوائية ذات الثماني خانات
  - ✓ لا يمكن التنبؤ بهذه السلسلة دون معرفة مفتاح الدخل
- يدعى خرج المولد بـ "سلسلة المفتاح" key stream
- يتم الربط بين كل بايت من بايتات هذا المفتاح وبايت واحد من سلسلة النص الصريح باستخدام العملية XOR

الفكرة الرئيسة لـ RC4(2/2)



التشفير



فك التشفير

آلية خوارزمية RC4 (1/3)

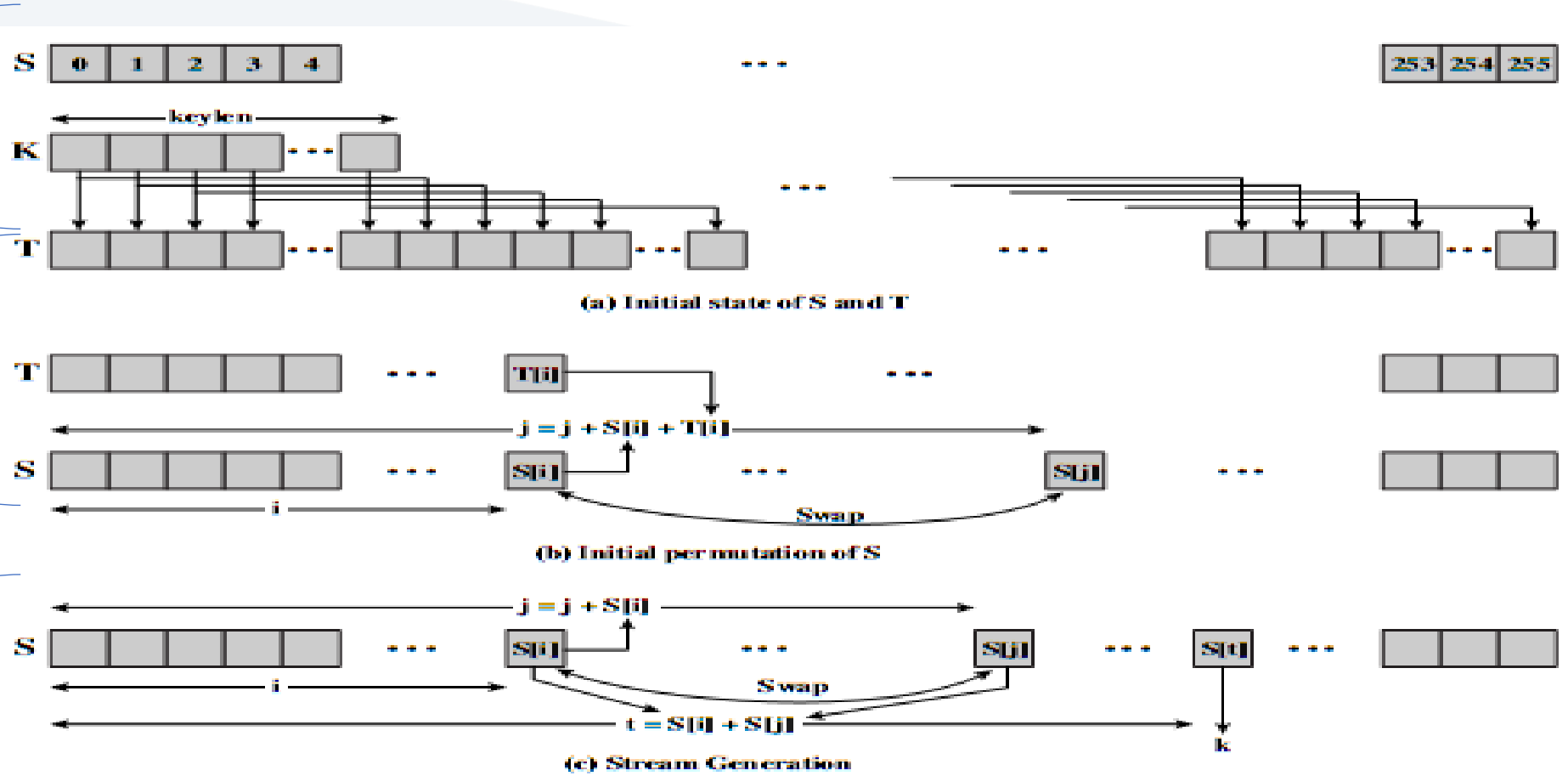
- يستخدم مفتاح ذو طول متغير bytes(1-256) (من 8 وحتى 2048 خانة) في تهيئة شعاع الحالة S المؤلف من256 bytes
- يضم شعاع الحالة S (مصفوفة سطرية)العناصر الآتية : $S[0], S[1], \ldots\ldots\ldots, S[255]$
- يحوي الشعاع S دائماً ترتيب تغيير المواضع لكل الأعداد ذات الثماني خانات من 0 وحتى 255
- يولد بايت المفتاح K المستخدم في عملية التعمية وفك التعمية من الشعاع S عن طريق اختيار واحد من 256 مدخلاً
- في كل مرة تولد فيها قيمة للمفتاح K، يتم تغيير المواضع في S
- تجرى عملية XOR بين بايت المفتاح وبايت النص الصريح المراد تشفيره.

آلية خوارزمية RC4 (3/3)

تهيئة الشعاع S و إنشاء الشعاع المؤقت T

تبديل المواضع الأولي للشعاع S

توليد سلسلة المفتاح



الخطوات التفصيلية لآلية عمل RC4 (1/4)

1. **بداية نهيء الشعاع S**: تأخذ كل مداخل الشعاع S قيماً مرتبة تصاعدياً من 0 وحتى 255، أي أن:

$$S[0] = 0, S[1] = 1, \ldots\ldots\ldots, S[255] = 255$$

2. **إنشاء شعاع مؤقت T**:

❖ بفرض مفتاح الدخل هو K وطوله هو keylen نميز حالتين:

✓ إذا كان (طول مفتاح الدخل =طول الشعاع S) أي keylen =256 bytes : عندها يوضع K كاملاً في T.

✓ إذا كان (طول مفتاح الدخل أصغر من طول الشعاع S) أي keylen < 256 bytes: عندها يُملأ أول keylen عنصراً من الشعاع T بعناصر المفتاح K، ومن ثم يكرر المفتاح K عدداً من المرات أو جزءاً منه فقط إلى أن يُملأ الشعاع T كاملاً

خوارزمية إنشاء شعاع مؤقت T

```
/* Initialization */
for  i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```

الخطوات التفصيلية لآلية عمل RC4 (2/4)

3. نستخدم بعد ذلك الشعاع T لإنتاج شعاع تبديل المواضع الأولي S.

تبدأ العملية بالعنصر S[0] والمتابعة حتى S[255]

فمن أجل كل S[i] : يبدل s[i] ببايت آخر S[j]من S

حيث يحسب موقع البايت (j) من العلاقة :

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

خوارزمية تبديل المواضع الأولي لـ S

```
/* Initial Permutation of S */
j = 0;
for  i = 0 to 255 do
j = (j + S[i] + T[i]) mod 256;
Swap (S[i], S[j]);
```

وبما أن العملية الوحيدة المطبقة على S هي التبديل، فإن التأثير الوحيد الناتج هو تبديل المواضع. لاحظ أن S مازال يحوي كل الأعداد من 0 وحتى 255.

الخطوات التفصيلية لآلية عمل RC4 (3/4)

4. توليد سلسلة المفتاح (Stream Generation)

```
/* Stream Generation */
i, j = 0;
while (true)
        i = (i + 1) mod 256;
        j = (j + S[i]) mod 256;
        Swap (S[i], S[j]);
        t = (S[i] + S[j]) mod 256;
        k = S[t];
```

- تنطلق عملية توليد السلسلة من الشعاع S الناتج عن عملية التهيئة
- يتضمن توليد السلسلة البدء بالعنصر S[0] ومن ثم المتابعة حتى S[255]، و من ثم يتم تبديل كل S[i] ببايت آخر من S حسب المخطط المفروض من التشكيل الحالي للشعاع S.
- بعد الوصول إلى S[255] تتم متابعة العملية بالبدء مرة أخرى بالعنصر S[0]

الخطوات التفصيلية لآلية عمل RC4 (4/4)

5. عملية التشفير (Encryption):

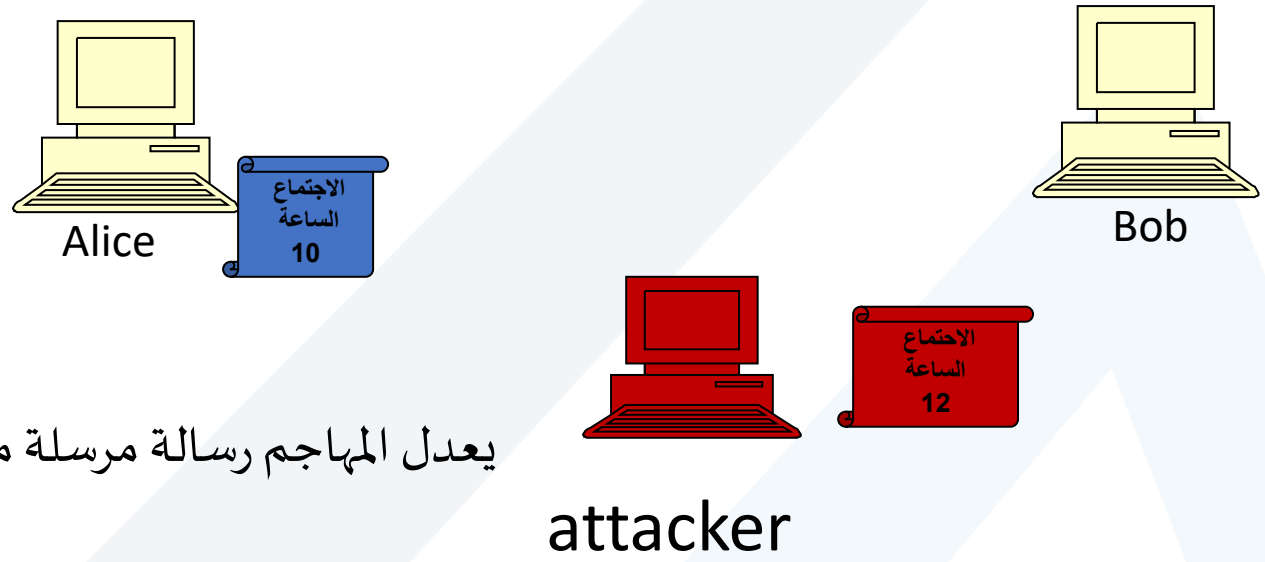
- تعتمد على تطبيق العملية XOR بين سلسلة المفتاح الناتج والبايت من النص الصريح.

6. عملية فك التشفير (Encryption):

- تعتمد على تطبيق عملية XOR بين سلسلة المفتاح الناتج والبايت المقابل من النص المشفر.

مقدمة إلى تابع البعثرة

- تعرف تكاملية المعطيات على أنها الخاصية التي تسمح بالتحقق من أن المعطيات لم تعدل من قبل كيان غير مخول له بذلك سواء بشكل مفاجئ أو مقصود.



يعدل المهاجم رسالة مرسلة من Alice إلى Bob

- يستخدم تابع البعثرة (hash) عملياً من أجل التحقق من متطلب تكاملية المعطيات.

تعريف تابع البعثرة (Hash function)

➢ **تعريفه:**

هو تابع يربط قناة ثنائية ذات طول متغير مع قناة ذات طول ثابت.

رياضياً هو تابع حسابي فعال يحول السلاسل الثنائية ذات الأطوال المتغيرة إلى سلاسل ثنائية ذات أطوال ثابتة (n). نرمز له بـ h():

$$h:\{0,1\}^{*}\rightarrow\{0,1\}^{n}, m\rightarrow h(m)$$

- ✓ مثالياً تكون قيم n ما بين 128-256 bits
- ✓ تدعى قيمة خرج تابع البعثرة بموجز الرسالة (message digest)
- ✓ يطلق عليه أحياناً : تابع البصمة (fingerprint function)

خصائص تابع البعثرة (Hash Function) (1/2)

1. **الضغط (Compression):** يجب أن ينتج خرج ذا طول ثابت وأصغر مقارنة مع أطوال الدخل.

Message	*Message Digest*
4523AB1352CDEF45126	13AB
723BAE38F2AB3457AC	02CA
AB45CD1048765412AAAB6662BE	A38B

2. **الفعالية(Efficiency):** يجب أن يكون سهل الحساب مهما كانت قيمة الدخل أي أن يكون من السهل حساب على قيمة h(m) من أجل قيمة دخل معلومة m

3. **وحيد الاتجاه (One-way):** من أجل قيمة y معطاة من غير الممكن إيجاد x بحيث $h(x) = y$

خصائص تابع البعثرة (Hash Function) (2/2)

4. **مقاوم للتصادمات (Strong collision resistance):** أي من أجل أية قيمتين مختلفتين ينتج حكماً خرجين مختلفين :

$$y \neq x \rightarrow h(y) \neq h(x)$$

❖ مثال 1: إذا كانت $Data\ X = (X_0, X_1, X_2, \ldots, X_{n-1})$ حيث X_i هي عبارة عن بايت

إذا فرضنا تابع بعثرة hash يعرف كالآتي: $h(X) = X_0 + X_1 + X_2 + \ldots + X_{n-1}$

هل هو آمن ؟

❖ الحل:

إذا كان لدينا $X = (10101010, 00001111)$ سيكون : $h(X) = 10111001$

لكن بفرض $Y = (00001111, 10101010)$ سيكون : $h(Y) = 10111001$

من السهل إيجاد دخلين مختلفين لهما نفس قيمة الخرج ، أي أنه غير مقاوم للتصادمات فهو تابع غير آمن.

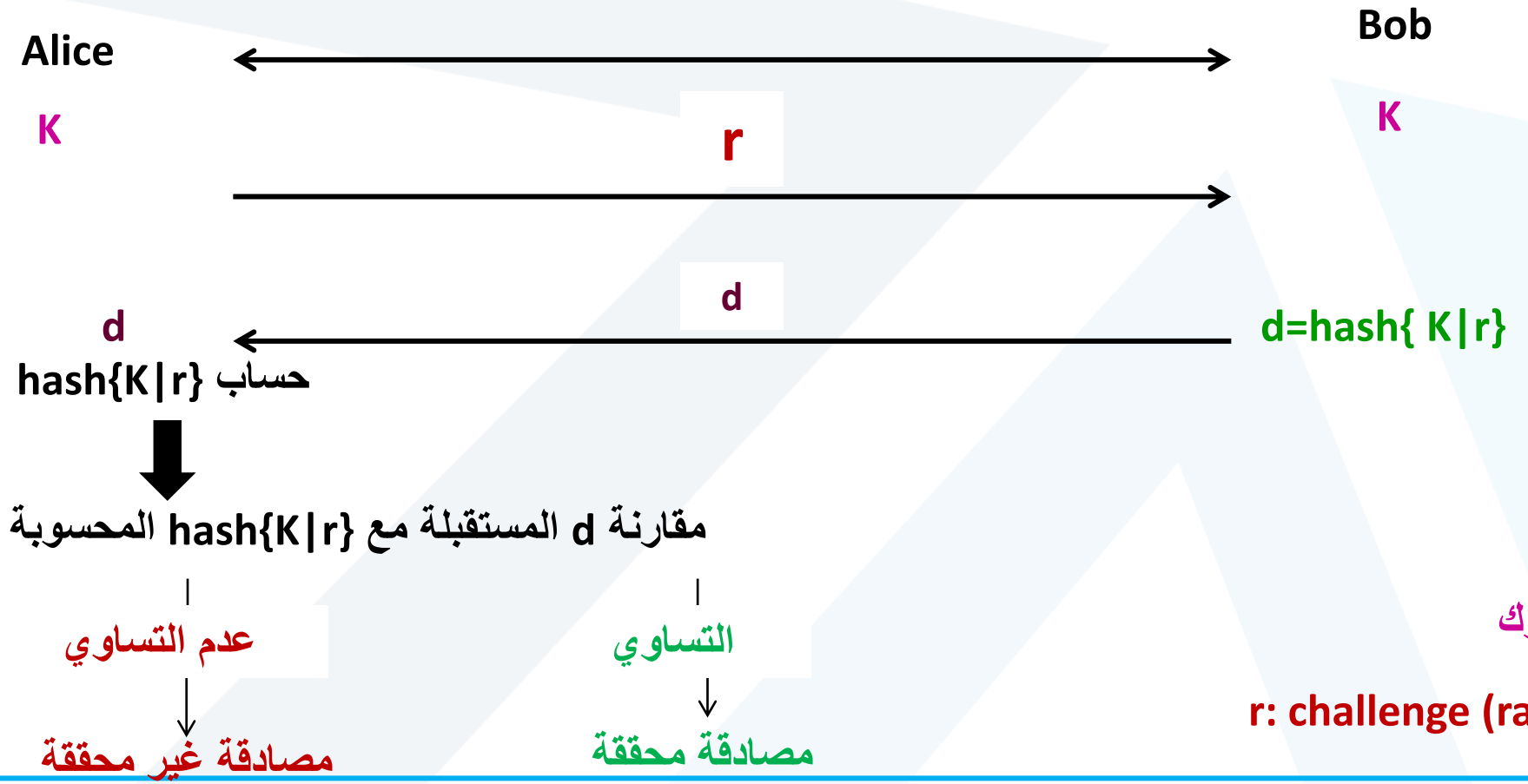
أهم استخدامات تابع البعثرة

1. تحقيق مصادقة أصل المعطيات
2. تحقيق متطلب المصادقة
3. تحقيق متطلب التكاملية

المصادقة باستخدام تابع البعثرة (1/2)

- في حال أراد طرفان التأكد من هوية كل منهما والتأكد من المفتاح المشترك بينهما
- يرسل أحدهما رقماً عشوائياً هذا الرقم العشوائي يكون بمثابة تحدي (challenge)
- الطرف الذي استقبل التحدي يطبق تابع البعثرة على قيمة هذه القيمة هي لصق لهذا التحدي مع المفتاح المشترك، و يرسل خرج تابع البعثرة الناتج إلى الطرف الآخر .
- عندما يستقبل ذلك الطرف خرج تابع البعثرة يعيد حساب تابع البعثرة باستخدام الرقم العشوائي و المفتاح المشترك ، و يقارن النتيجة مع تابع البعثرة التي استقبلها .
- هنا نميز حالتين:
  - ✓ في حال التساوي تكون المصادقة محققة
  - ✓ في حال عدم التساوي تكون المصادقة غير محققة

المصادقة باستخدام تابع البعثرة (2/2)



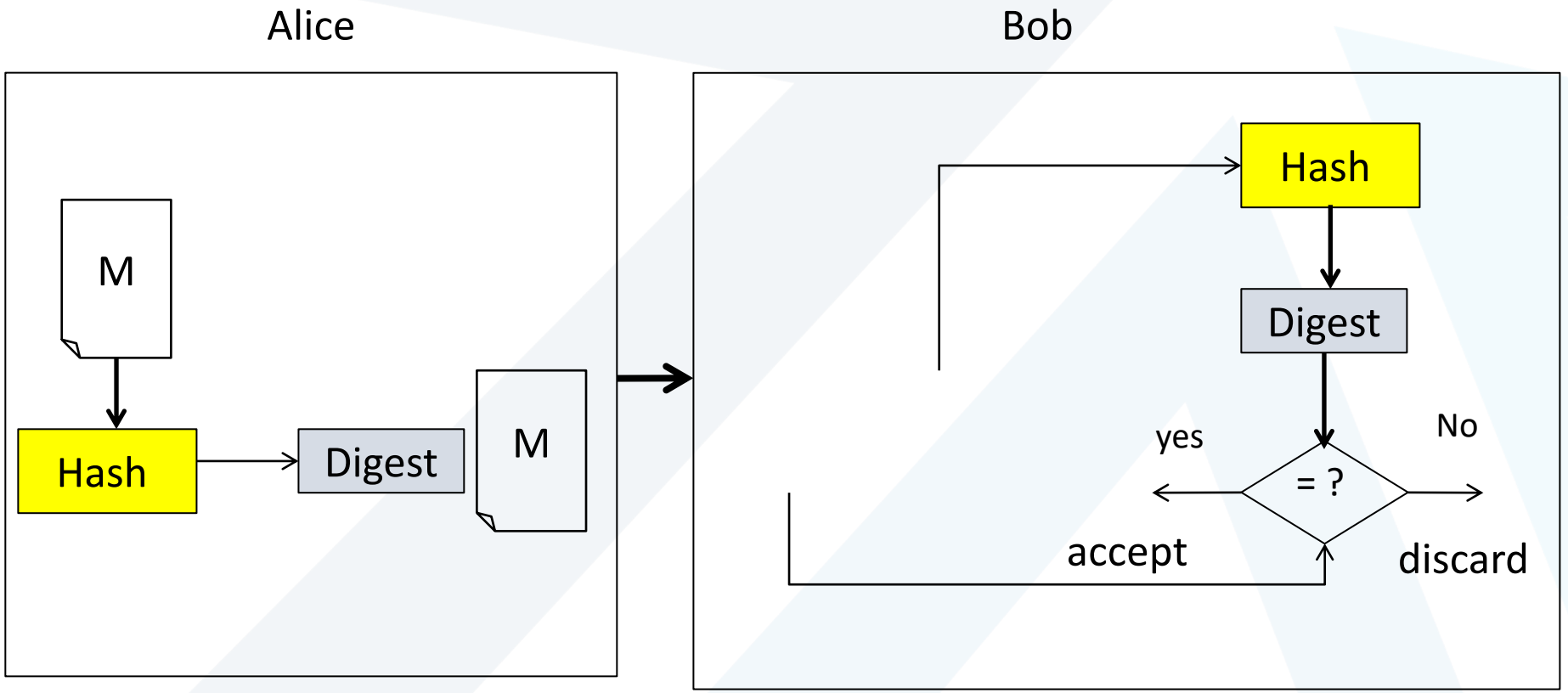
مفتاح مشترك: K

r: challenge (random number)

تكاملية البيانات باستخدام تابع البعثرة (1/2)

- الغاية من استخدام تابع البعثرة هنا التأكد من أن الرسالة الأصلية لم تعدل، ولكن لا يضمن التحقق من أصل المرسل
- يطبق المرسل تابع البعثرة على الرسالة و من ثم يرسل خرج التابع ملصقاً مع الرسالة
- في جهة الاستقبال يحسب المستقبل تابع البعثرة للرسالة الواصلة إليه ويقارن الناتج مع قيمة تابع البعثرة المستقبلة
  - في حال التطابق الرسالة صحيحة لم تتعرض للتعديل ومتطلب التكاملية محقق
  - في حال عدم التطابق الرسالة معدلة و متطلب التكاملية غير محقق

تكاملية البيانات باستخدام تابع البعثرة (2/2)



تصنيف توابع البعثرة (1/2)

تصنف إلى نوعين أساسيين:

- **توابع بعثرة دون مفتاح :**

هي توابع البعثرة التي تمتلك دخل واحد هو الرسالة .

مثال عنها **MDC (Manipulation Detection Code)**

- في MDC : خرج تابع البعثرة (H(M))يلصق مع المعلومات الأصلية التي طبق التابع عليها(M) ومن ثم تشفر بهدف الحماية من العبث فيها. أي يُرسل في قناة موثوقة كالآتي:

$$MDC = E_K(M \| H(M))$$

حيث K هو المفتاح السري.

تصنيف توابع البعثرة (2/2)

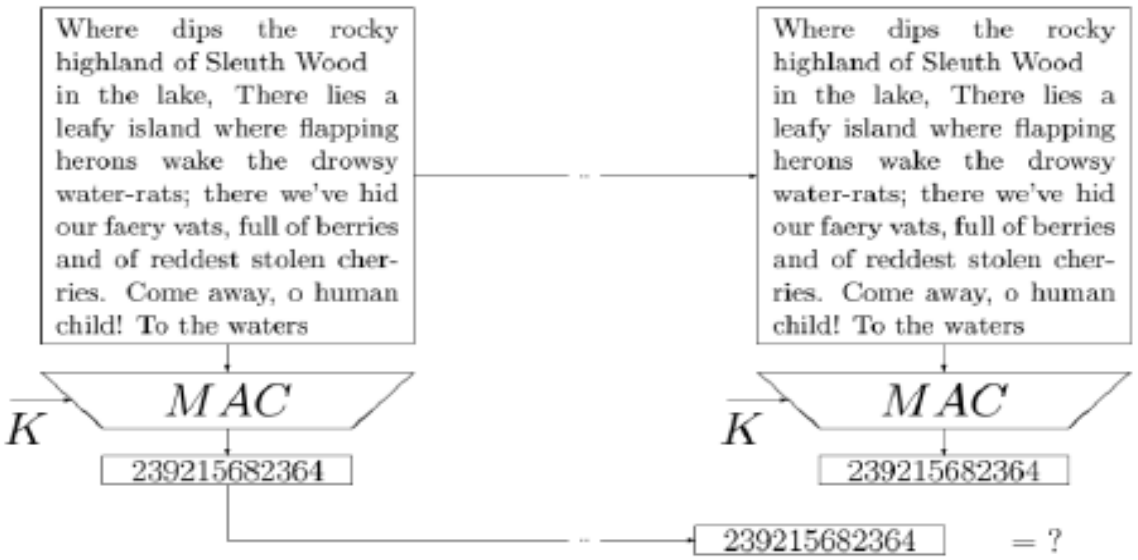
تصنف إلى نوعين أساسيين:

- **توابع بعثرة مع مفتاح :**

هي توابع البعثرة التي تمتلك دخلين اثنين أحد هما هو المفتاح السري و الآخر هو الرسالة.

مثال عنها MAC(Message Authentication Code)

تحسب الـ MAC و يرسل الآتي: $M \| MAC_K(M)$



Message Authentication Code (MAC)

- **آلية العمل:**

✓ يتشارك المرسل A والمستقبل B بنفس المفتاح السري

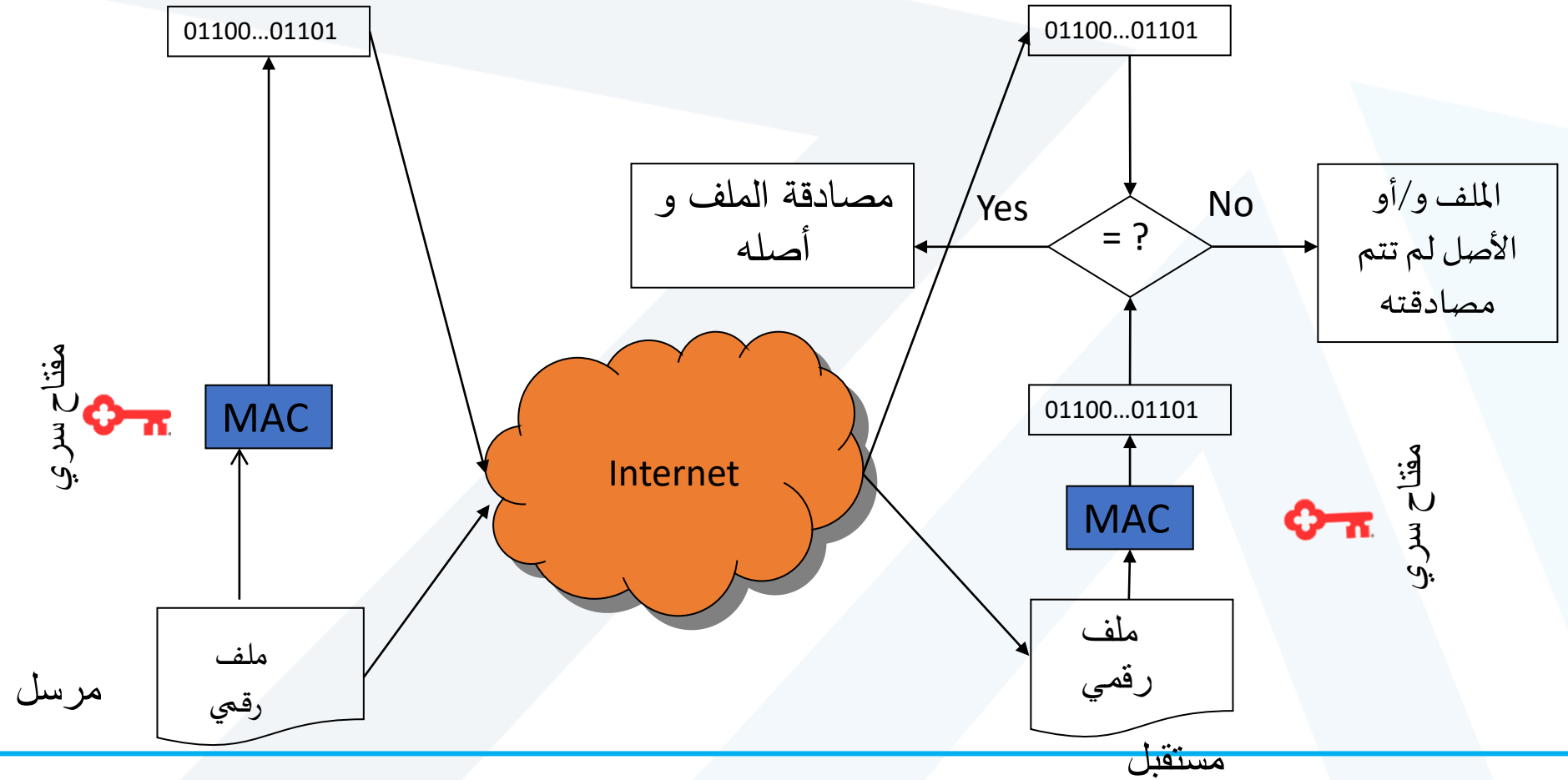
✓ يحسب المرسل $MAC_{K_{AB}}(m)$ و تلصق مع الرسالة (m) من أجل إرسالها باستخدام المفتاح السري K_{AB}

✓ يرسل المرسل $m \| MAC_{K_{AB}}(m)$

✓ بعد استقبال الرسالة، يبحث المستقبل عن مصدر الرسالة المستقبلة كما يلي:

- يعيد المستقبل حساب الـ $MAC_{K_{AB}}(m)$ للرسالة المستقبلة باستخدام المفتاح السري K_{AB}
- يقارن هذه النتيجة مع الـ $MAC_{K_{AB}}(m)$ المستقبلة ويحدد فيما كانت الرسالة المستقبلة أصلية و مرسلة من المصدر الصحيح أم لا

مصادقة أصل المعطيات باستخدام الـ MAC



أمثلة عن تابع البعثرة (Hash Function)

- MD2 (Message Digest 2)
- MD5 (Message Digest 5)
- SHA-1 (Secure Hash Algorithm 1)

Hash Algorithm	Digest Size (بت)	سرعة الحساب (ميغابايت/ثانية)
MD5	128	204.55
SHA-1	160	72.60

على جهاز مع معالج Pentium 4، 2.1 GHz وتحت Win XP SP1

- مثال عن الـ MAC هو الخوارزمية HMAC/MD5

MAC Algorithm	Digest Size (بت)	سرعة الحساب (ميغابايت/ثانية)
HMAC/MD5	128	215.76

جَامعة
المَنارة
MANARA UNIVERSITY

نهاية المحاضرة السابعة