



Information System Security أمن نظم المعلومات

مدرسة المقرر

د. بشرى علي معلا

الأربعاء 6/7/2023

العام الدراسي 2022-2023

<https://manara.edu.sy/>

الفصل الدراسي الثاني



عناوين المحاضرة السابعة

➤ مقدمة

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

<https://manara.edu.sy/>



مقدمة

• تقسم خوارزميات التشفير المتناظر إلى :

✓ خوارزميات التشفير الكتلّي: 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:

10000101010

11001010101 ⊕

01001111111

■ النص الصريح: 10000101010

■ سلسلة المفتاح: 11001010101

<https://manara.edu.sy/>

3



تعريف بخوارزمية RC4 واستخداماتها

➤ صممت هذه الخوارزمية في عام 1987 من قبل Ron Rivest

➤ تعد RC4 نظام التعمية التسلسلي الأكثر انتشاراً

➤ تستخدم RC4 طول مفتاح متغير، وتطبق 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

➤ استخدامها:

✓ في البروتوكول SSL/TLS المستخدم بين مستكشفات الويب

❖ WEP (Wired Equivalent Privacy)

✓ في المعيار IEEE 802.11 wireless LAN

❖ WPA (WiFi Protocol Access)

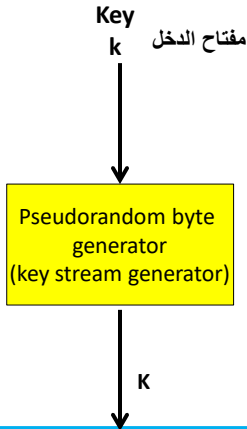
✓ ضمن البروتوكولين:

<https://manara.edu.sy/>

4



الفكرة الرئيسية لـ RC4 (1/2)

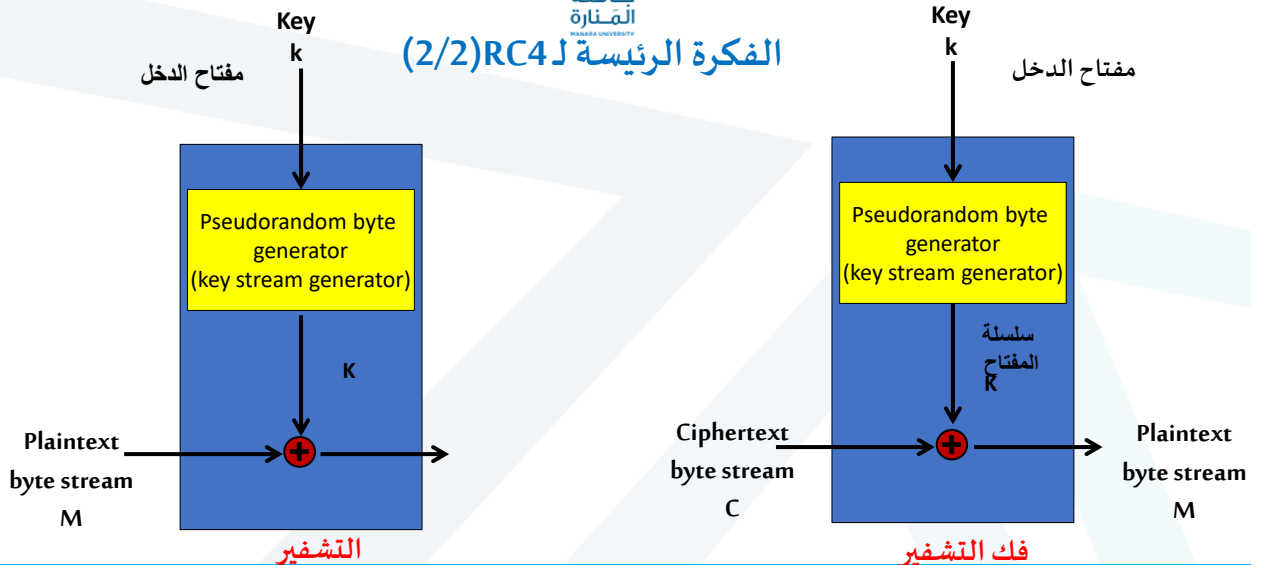


- يعتمد على تعمية النصوص الصريحة على شكل بايت واحد في كل عملية (بنفس الوقت)
- يمثل المفتاح في هذه البنية دخل مولد خانات شبه عشوائي Pseudorandom byte generator
 - ✓ ينتج المولد سلسلة من الأعداد العشوائية ذات الثماني خانات
 - ✓ لا يمكن التنبؤ بهذه السلسلة دون معرفة مفتاح الدخل
- يدعى خرج المولد بـ "سلسلة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

5

<https://manara.edu.sy/>


الفكرة الرئيسية لـ RC4 (2/2)



6

<https://manara.edu.sy/>



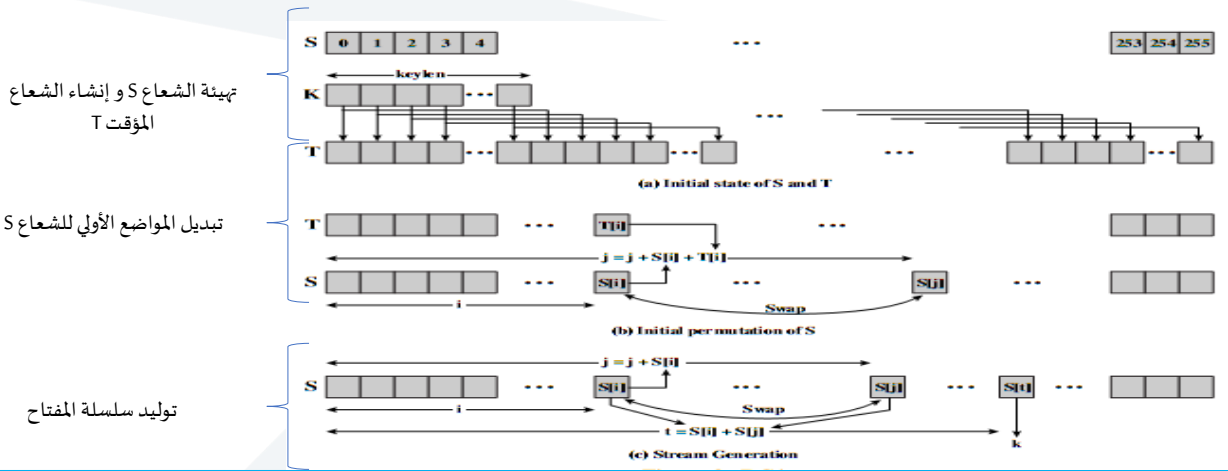
آلية خوارزمية RC4 (1/3)

- يستخدم مفتاح ذو طول متغير bytes (1-256) (من 8 وحتى 2048 خانة) في تهيئة شعاع الحالة S المؤلف من 256 bytes
- يضم شعاع الحالة S (مصفوفة سطرية) العناصر الآتية: $S[0], S[1], \dots, S[255]$
- 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.

7

<https://manara.edu.sy/>


آلية خوارزمية RC4 (3/3)



8

<https://manara.edu.sy/>



الخطوات التفصيلية لألية عمل RC4 (1/4)

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

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

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];
```

9

<https://manara.edu.sy/>



الخطوات التفصيلية لألية عمل RC4 (2/4)

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

خوارزمية تبديل المواضع الأولي 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]);
```

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 البايـت (j) من العلاقة :

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

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.

10

<https://manara.edu.sy/>



الخطوات التفصيلية لألية 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 لألية عمل RC4 (4/4)

5. عملية التشفير (Encryption):

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

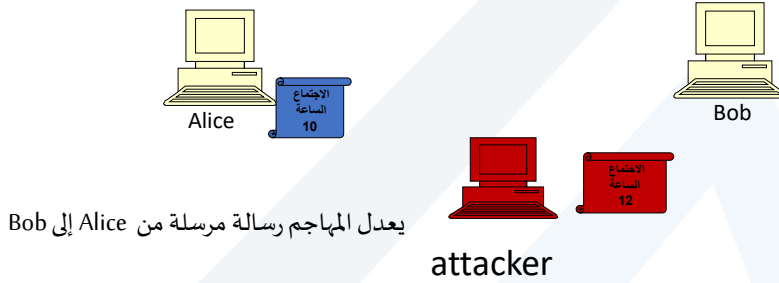
6. عملية فك التشفير (Decryption):

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



مقدمة إلى تابع البعثة

➤ 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.



➤ يستخدم تابع البعثة (hash) عملياً من أجل التحقق من متطلب تكاملية المعطيات.

13

<https://manara.edu.sy/>


تعريف تابع البعثة (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). نرسم 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 البعثة بموجز الرسالة (message digest)
- ✓ يطلق عليه أحياناً: تابع البصمة (fingerprint function)

14

<https://manara.edu.sy/>



خصائص تابع البعثة (Hash Function) (1/4)

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 $h(x) = y$

<https://manara.edu.sy/>

15



خصائص تابع البعثة (Hash Function) (2/4)

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

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

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

$$h(X) = X_0 + X_1 + X_2 + \dots + X_{n-1}$$

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.

<https://manara.edu.sy/>

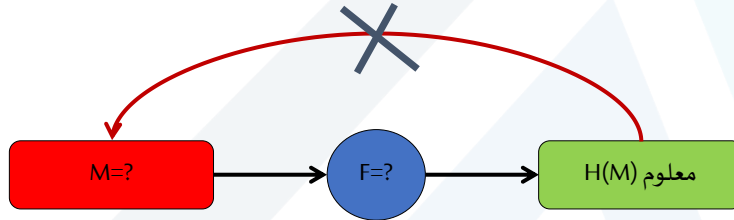
16



خصائص تابع البعثة (Hash Function) (3/4)

5. مقاوم ضد هجوم الصورة الأولية (Pre-image Attack)

➤ إذا كانت قيمة التابع $h(m)$ معروفة للمهاجم فإنه من الصعب حساب m



هذه الخاصية تحمي ضد المهاجم الذي يمتلك قيمة تابع البعثة ويحاول إيجاد قيمة الدخل

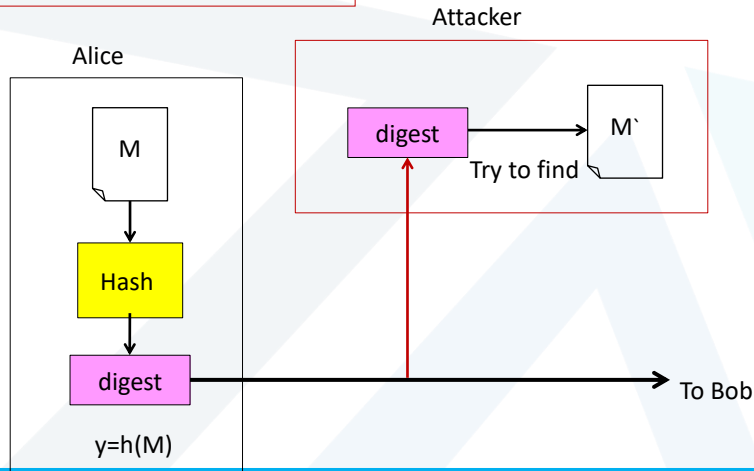
<https://manara.edu.sy/>

17



Preimage Attack

m' معلومة يحاول إيجاد $h(m)$



<https://manara.edu.sy/>

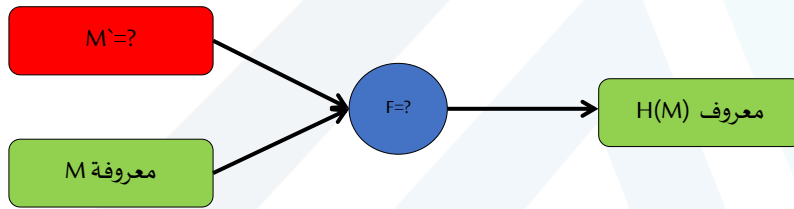
18



خصائص تابع البعثة (Hash Function) (4/4)

6. مقاوم ضد هجوم الصورة الثانية (Second pre-image Attack)

- من أجل قيم رسالة m و تابع البعثة $h(m)$ معروفة من قبل المهاجم ، فإنه من الصعب إيجاد رسالة أخرى m' بحيث يكون لها نفس قيمة تابع البعثة: أي من الصعب جداً تحقيق: $hash(m)=hash(m')$



هذه الخاصية تحمي من المهاجم الذي يمتلك الدخل وقيمة خرج تابع البعثة الموافق له و يريد أن يستبدل القيمة الأصلية بقيمة مختلفة كقيمة شرعية

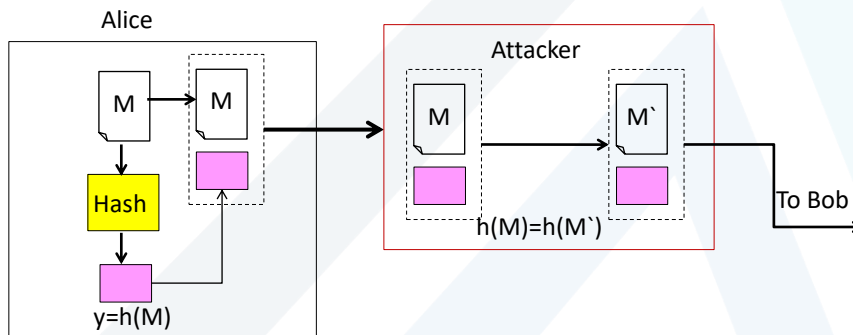
<https://manara.edu.sy/>

19



Pre-image Attack Second

m مختلفة عن m' معلومة يحاول إيجاد $h(m)$, $h(m')$ بحيث $h(m)=h(m')$



<https://manara.edu.sy/>

20



أهم استخدامات تابع البعثة

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

<https://manara.edu.sy/>

21



المصادقة باستخدام تابع البعثة (1/2)

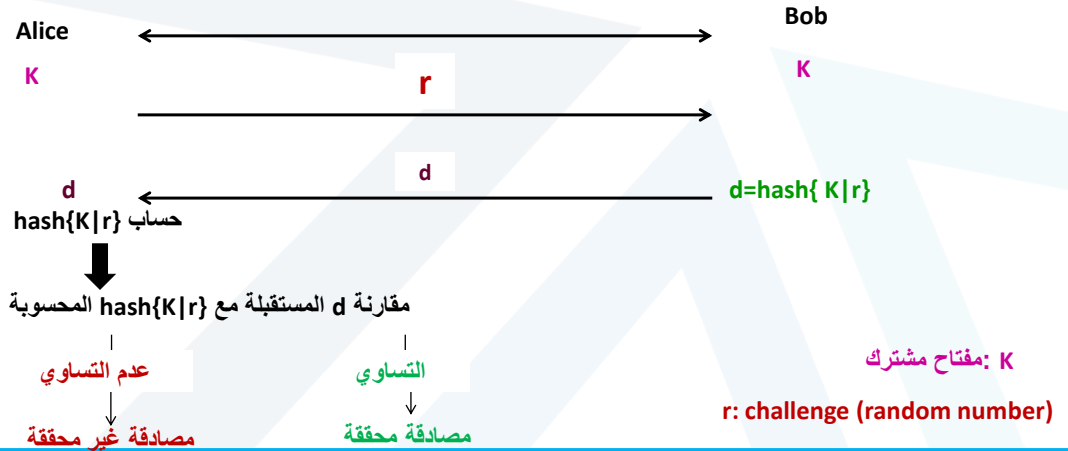
- في حال أراد طرفان التأكد من هوية كل منهما والتأكد من المفتاح المشترك بينهما
- يرسل أحدهما رقماً عشوائياً هذا الرقم العشوائي يكون بمثابة تحدي (challenge)
- الطرف الذي استقبل التحدي يطبق تابع البعثة على قيمة هذه القيمة هي لصق لهذا التحدي مع المفتاح المشترك، و يرسل خرج تابع البعثة الناتج إلى الطرف الآخر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

<https://manara.edu.sy/>

22



المصادقة باستخدام تابع البعثة (2/2)



23

<https://manara.edu.sy/>


تكاملية البيانات باستخدام تابع البعثة (1/2)

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

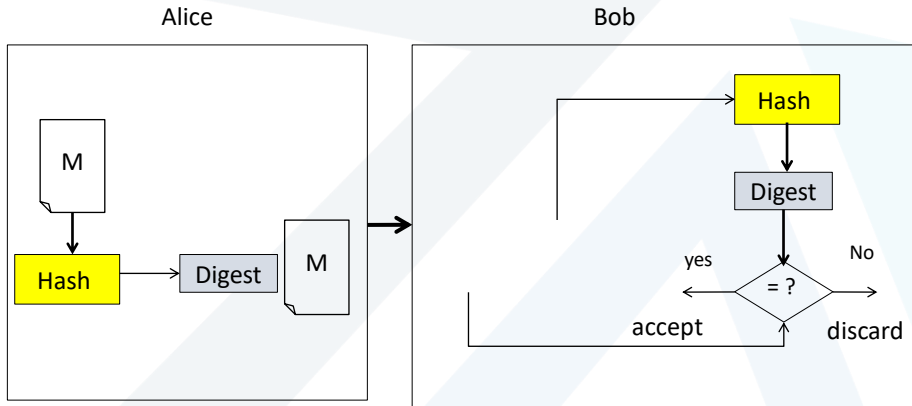
24

<https://manara.edu.sy/>



جامعة
المنارة
MANARA UNIVERSITY

تكاملية البيانات باستخدام تابع البعثة (2/2)



25

<https://manara.edu.sy/>



جامعة
المنارة
MANARA UNIVERSITY

تصنيف توابع البعثة (1/2)

تصنف إلى نوعين أساسيين:

❖ توابع بعثة دون مفتاح :

هي توابع البعثة التي تمتلك **دخول واحد** هو الرسالة .

مثال عنها (MDC (Manipulation Detection Code

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

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

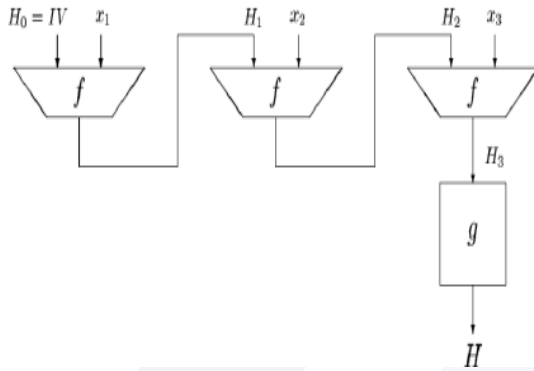
حيث K هو المفتاح السري.

26

<https://manara.edu.sy/>



■ أغلب توابع الـ MDC مبنية على تكرار تابع البعثرة وتقسيم الرسالة إلى بلوكات



$h_0 = IV = \text{a fixed initial value}$

$h_1 = H(h_0, M_1)$

\vdots

$h_i = H(h_{i-1}, M_i)$

$h_n = H(h_{n-1}, M_n)$

$h(M) = h_n$

حيث: F هو تابع البعثرة. g هو تابع تحويل.

27

<https://manara.edu.sy/>



تصنيف توابع البعثرة (2/2)

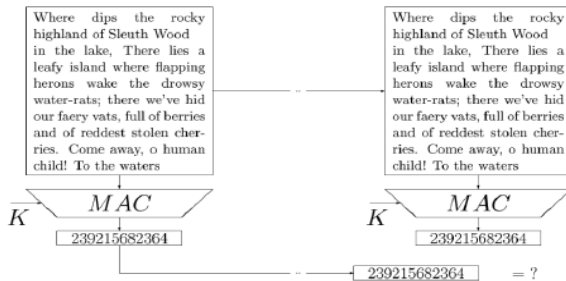
تصنف إلى نوعين أساسيين:

❖ توابع بعثرة مع مفتاح:

هي توابع البعثرة التي تمتلك **دخلين اثنين** أحدهما هو المفتاح السري والآخر هو الرسالة.

مثال عنها MAC (Message Authentication Code)

تحسب الـ MAC ويرسل الآتي: $M || MAC_K(M)$



28

<https://manara.edu.sy/>



Message Authentication Code (MAC)

آلية العمل:

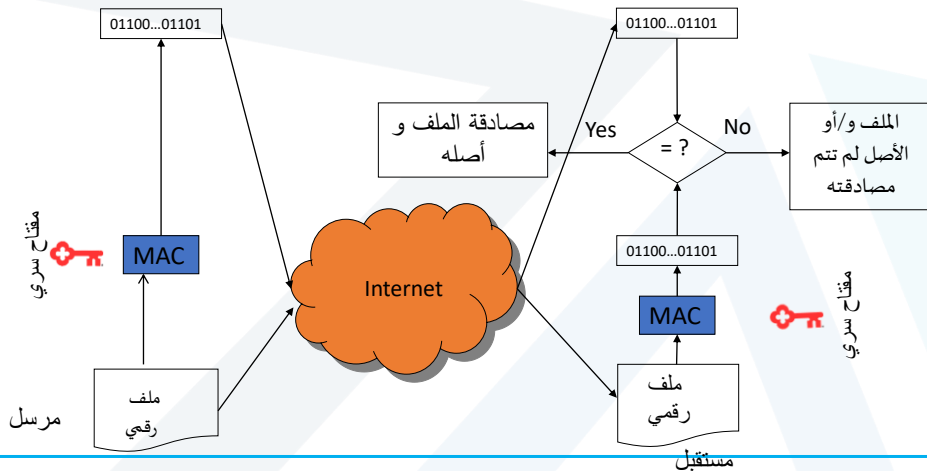
- ✓ يتشارك المرسل A والمستقبل B بنفس المفتاح السري
- ✓ يحسب المرسل $MAC_{K_{AB}}(m)$ و تلتصق مع الرسالة (m) من أجل إرسالها باستخدام المفتاح السري K_{AB}
- ✓ يرسل المرسل $m || MAC_{K_{AB}}(m)$
- ✓ بعد استقبال الرسالة، يبحث المستقبل عن مصدر الرسالة المستقبلية كما يلي:
 - يعيد المستقبل حساب الـ $MAC_{K_{AB}}(m)$ للرسالة المستقبلية باستخدام المفتاح السري K_{AB}
 - يقارن هذه النتيجة مع الـ $MAC_{K_{AB}}(m)$ المستقبلية ويحدد فيما كانت الرسالة المستقبلية أصلية و مرسله من المصدر الصحيح أم لا

29

<https://manara.edu.sy/>



مصادقة أصل المعطيات باستخدام الـ MAC



30

<https://manara.edu.sy/>



أمثلة عن تابع البعثة (Hash Function)

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

MD2 (Message Digest 2) ➤

MD5 (Message Digest 5) ➤

SHA-1 (Secure Hash Algorithm 1) ➤

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

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

➤ مثال عن الـ MAC هو الخوارزمية HMAC/MD5

31

<https://manara.edu.sy/>



نهاية المحاضرة السابعة

<https://manara.edu.sy/>