# الفصل–7 التوقيع الرقمي

## 7.1 المقدمة



- في هذا الفصل ، سوف ندرس مناهج التوقيع الرقمية.
- يعتبر التوقيع الرقمي الوسيلة التي نوقّع بها رسالة مخزونة بشكل الكتروني ، بحيث يمكن تناقلها عبر شبكة الحاسوب.
- يختلف التوقيع الرقمي عن التوقيع التقليدي بكون التوقيع فيه منفصلا عن الرسالة الموقّعة ، وكونه صعب التزوير forge ، لانه يمكن استخدام خوارزمية فحص معلنة للتأكد من صلاحية التوقيع الرقمي.
- هناك اختلاف اخر بين التواقيع الرقمية والتقليدية وهو ان "نسخ" الرسالة الموقّعة الكترونيا تكون مطابقة للرسالة الاصلية ، في حين ان نسخة الرسالة الموقعة ورقيا يمكن تمييزها بسهولة عن الاصلية ، مما يستدعي اخذ الحذر من سوء استخدام التوقيع الرقمي.
  - على سبيل المثال ، عندما توقّع Alice رسالة رقمية تسمح للشخص Bob بسحب مبلغ 100 دولار من حسابها المصرفي ، فانها تطمح ، بنفس الوقت ، ان تمنع Bob من تكرار عملية السحب اكثر من مرة. لذا يجب ان تتضمن الرسالة معلومات مثل الوقت والتاريخ لمنع استخدامها اكثر من مرة.
- تسمح التواقيع الرقمي للطرف **الموقّع** signer ، الذي لديه مفتاح معلن $pk$ ، ان يوقّع sign رسالة ما بحيث ان اي طرف اخر يعرف هذا المفتاح يستطيع **اختبار** verify ان هذه الرسالة موقّعة فعلا من قبل صاحب ذلك المفتاح ويتأكد ايضا من ان الرسالة لم تتغير اثناء الارسال. يمكن اعتبار التوقيع الرقمي على انه نظير شفرات توثيق الرسالة MAC ولكن في بيئة المفتاح المعلن.
- احد الامثلة على استخدام التوقيع الرقمي هو الشركات المنتجة للبرامجيات التي ترغب بمنع انتشار البرامجيات غير الموثّقة. حيث تقوم الشركة بتوقيع اصداراتها بمفتاح معلن ، بحيث ان اي زبون يستطيع التأكد من وثوقية الاصدار الذي لديه ، وبذلك تمنع الشركة اي طرف اخر من استغفال الزبائن بنسخة لم تصدر من تلك الشركة.
  - لفعل هذا ، تقوم الشركة بتوليد مفتاح معلن $pk$ مع مفتاح خاص $sk$ ، وتقوم الشركة بتوزيع المفتاح المعلن لزبائنها بينما تحتفظ بسرية المفتاح الخاص. عند اصدار منتج جديد $m$ ، تقوم الشركة بحساب **التوقيع الرقمي** $\sigma$ للمنتج $m$ باستخدام المفتاح $sk$. يتم ارسال الزوج $(m, \sigma)$ لجميع الزبائن. يستطيع كل زبون ان يختبر كون $\sigma$ هو توقيع رقمي قانوني للرسالة $m$ باستخدام المفتاح المعلن $pk$.
  - لاحظ ، ان نفس المفتاح المعلن يستخدم من قبل الشركة لتوليد توقيع واحد فقط لجميع الزبائن.
  - يحاول الطرف المخرّب ان يصدر منتج مزيف وذلك بارسال $(m', \sigma')$ لاحد الزبائن ، حيث ان $m'$ هو منتج لم يتم اصداره مطلقا من قبل الشركة. اذا كان منهج التوقيع "آمنا" فإن فحص الزبون للتوقيع $\sigma'$ سوف ينتج ان ذلك التوقيع غير صالح بالنسبة للمنتج $m'$ عند استخدام $pk$.

### 7.1.1 مقارنة مع شفرات توثيق الرسالة

- يستخدم كلا من التوقيع الرقمي وشفرات توثيق الرسالة MAC لضمان سلامة (او توثيق) الرسائل المرسلة.
- يُبّسط استخدام التوقيع الرقمي مسألة ادارة المفاتيح في حالة حاجة المرسل للتواصل مع عدة مستلمين.
  - بمعنى اخر ، يتجنب المرسل ، في حالة التوقيع الرقمي ، تأسيس مفتاح مختلف مع كل مستلم محتمل ، ويتجنب حساب شفرة توثيق مختلفة لكل مفتاح. بدلا من ذلك ، يحسب المرسل التوقيع الرقمي مرة واحدة للرسالة ويفحص هذا التوقيع من قبل جميع المستلمين.

- الحسنة الاساسية للتوقيع الرقمي مقارنة مع شفرات توثيق الرسالة هي ان التوقيع الرقمي قابل **للفحص المعلن** publicly verifiable. بمعنى لو قام احد المستلمين بفحص صحة التوقيع الرقمي فإن بقية المستلمون سيتأكدون من صحة ذلك التوقيع. اما في حالة شفرة توثيق الرسالة

MAC فلو قام الطرف المخرّب بحساب شفرة MAC غير صحيحة لاحد المستلمين وبمفتاح ذلك المستلم فانه سوف يرفض الرسالة المرسلة بدون ان يتأكد من رفض او قبول بقية المستلمين وذلك لكون المفاتيح مستقلة.

- يمتاز التوقيع الرقمي ايضا بكونه **قابلا للنقل** transferable: يمكن ان نظهر التوقيع الرقمي $\sigma$ والرسالة $m$ الى طرف ثالث ، لكي يثبت صحة التوقيع باستخدام مفتاح الموقّع المعلن $pk$. كما يمكن ان يرسل الطرف الثالث نسخة التوقيع الى طرف اخر ، ليحكم بصلاحية التوقيع.



- توفّر التواقيع الرقمية ميزة مهمة وهي **عدم الانكار** non-repudiation. بمعنى ان موقّع الرسالة لايستطيع انكار توقيعه على الرسالة. وهذه ميزة حاسمة بالنسبة للمستلمين الذين يحتاجون الى برهنة كون مرسل الرسالة قد وقعّها بمفتاحه الخاص: على افتراض توفر مفتاح الموقّع المعلن لدى القاضي ، فإن صلاحية التوقيع هي دليل بحد ذاته على ان ذلك الموقّع هو فعلا من قام بتوقيع تلك الرسالة.
  - لاتمتلك شفرات توثيق الرسالة هذه الميزة. لنرى ذلك ، نفترض ان المرسل والمستلم يتّفقون على مفتاح خاص $ksr$. يقوم المرسل بارسال رسالة مع شفرة توثيق الى المستلم. بما ان القاضي لايعرف المفتاح الخاص $ksr$ فلا توجد طريقة يستطيع ان يثبت بها القاضي عائدية الرسالة للمرسل. عندما يكشف المشتكي المفتاح $ksr$ الى القاضي ، فلا يمكن للقاضي التأكد من ذلك ، لكون ذلك المفتاح مشترك بين الطرفين.
- كما هو حال التشفير معلن المفتاح ، تملك شفرات توثيق الرسالة ميزة كونها اسرع بمقدار3-2 مرات من التوقيع الرقمي ، وبذلك يفضّل استخدام شفرات توثيق الرسالة في الحالات التي لاتتطلب فحص معلن ، نقل التوقيع ، او عدم الانكار.

## 7.2 تعريف التوقيع الرقمي

- تعّرف الخوارزمية التي يطبّقها المرسل على الرسالة بخوارزمية $Sign$ ، ويسمى اخراج تلك الخوارزمية بالتوقيع signature.
- تدعى الخوارزمية التي يطبّقها المستلم على الرسالة والتوقيع لفحص صلاحية التوقيع بخوارزمية $Vrfy$. نقوم الأن بتعريف نظام التوقيع الرقمي بصورة شكلية.

**تعريف**(7.1): يتكون **منهج التوقيع الرقمي** من ثلاث خوارزميات احتمالية $(Gen, Sign, Vefy)$ وتحقق مايلي:

1. تأخذ خوارزمية توليد المفتاح ، $\mathbf{Gen}$ ، معامل الأمنية $n$ وتخرج زوج من المفاتيح $(pk, sk)$. وهي المفتاح المعلن والمفتاح الخاص ، على التوالي. نفترض ان كلا المفتاحين لهما نفس الطول $n$.
2. تأخذ خوارزمية التوقيع ، $\mathbf{Sign}$ ، المفتاح الخاص $sk$ والرسالة $m$ من فضاء رسائل معين. تخرج التوقيع $\sigma$ ، ونكتب هذا بالشكل $\sigma \leftarrow Sign_{sk}(m)$.
3. تأخذ خوارزمية الفحص ، $\mathbf{Vrfy}$ ، المفتاح المعلن $pk$ ، الرسالة $m$ ، والتوقيع $\sigma$ كمدخلات

- وتخرج الثنائية $b$ ، تعني $b = 1$ ان التوقيع صالح valid
- بينما تعني $b = 0$ ان التوقيع غير صالح invalid.

يكون اخراج الخوارزمية محدّد ويكتب بالشكل $b = Vrfy_{pk}(m, \sigma)$.

يشترط ان يكون لجميع قيم $n$ ، كل زوج $(pk, sk)$ ناتج من $Gen$ ، وكل رسالة $m$ ، ان يكون

$$Vrfy_{pk}\big(m, Sign_{sk}(m)\big) = 1$$

يستخدم التوقيع الرقمي بالطريقة التالية:

- يقوم المرسل بتنفيذ خوارزمية $Gen(n)$ للحصول على المفاتيح $(pk, sk)$.

- يتم الاعلان عن المفتاح المعلن على انه تابع الى المرسل $S$.
- عندما يريد $S$ نقل الرسالة $m$ الى طرف معين اوعدة اطراف يقوم $S$ بحساب التوقيع الرقمي $\sigma = \mathrm{Sign}_{sk}(m)$ ويرسل $(m, \sigma)$.
- بعد ان يستلم $(m, \sigma)$ ، يقوم المستلم الذي يعرف $pk$ بفحص وثوقية $m$ بفحص فيما اذا كانت $\mathrm{Vrfy}_{pk}(m, \sigma) = 1$. يثبت هذا الاجراء ان $S$ ارسل $m$ وايضا ان $m$ لم تتغير عند الارسال. ولكون التوقيع لايتضمن وقت ارسال الرسال ، فانه يمكن ان يتعرض لهجمات replay-attacks.



## 7.3 أمنية مناهج التوقيع الرقمي

في هذا الجزء سوف نحدّد معنى كون منهج التوقيع "آمنا". لذا سوف نبتدأ بوصف انواع المهاجمات ، واهداف الخصم المحتملة.

تختلف انواع المهاجمات اعتمادا على كمية المعلومات المتوفرة لدى الخصم. يكون هناك عدة انواع معتبرة من المهاجمات في حالة التوقيع الرقمي:

1. مهاجمة المفتاح فقط key-only attack: يمتلك الخصم المفتاح المعلن للموقّع ، وبالتالي يستطيع ان يفحص التوقيع فقط.
2. مهاجمة الرسالة المعلومة فقط known-message attack: يمتلك الخصم قائمة من الرسائل الموقّعة ، $(m_1, \sigma_1), (m_2, \sigma_2), \ldots$.
3. مهاجمة الرسالة المختارة Chosen message attack(**CMA**): يجهّز الخصم قائمة من الرسائل $(m_1, m_2, \ldots)$ ويطلب من الموقّع ان يوقعها له.

في ضوء المعلومات المتوفرة يهدف الخصم عدة امور محتملة:

1. التهديد الكامل Total break. يحاول الخصم معرفة المفتاح الخاص للموقّع وبالتالي يحسب التوقيع لأي رسالة.
2. التزوير الانتقائي Selective forgery . يحاول الخصم ان يحسب التوقيع لرسالة تابعة لشخص اخر ولم توقّع من قبل.
3. التزوير الوجودي Existential forgery . عند اعطاء المفتاح المعلن $pk$ المولّد من قبل $S$ ، نقول ان الخصم يخرج تزوير forgery اذا اخرج رسالة $m$ مع توقيع صالح $\sigma$ من $m$ ، وان تلك الرسالة لم تكن موقّعة مسبقا من قبل $S$.

- تعني أمنية منهج التوقيع الرقمي ان الخصم لايستطيع اخراج اي تزوير رغم وصوله الى خوارزمية التوقيع $\mathrm{Sign}$ وحصوله على تواقيع لرسائل متعددة من اختياره.

ليكن $\pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Vrfy})$ منهج توقيع ، ولتكن لدينا التجربة التالية للخصم $\mathcal{A}$ ومعامل الأمنية $n$.

**تجربة التوقيع $\mathbf{Sig - forge}^{\mathbf{cma}}_{\mathcal{A},\pi}(\mathbf{n})$:**

1. تنفيذ $\mathrm{Gen}(n)$ للحصول على المفاتيح $(pk, sk)$.
2. يعطى الخصم $\mathcal{A}$ المفتاح المعلن $pk$ وحق الوصول الى $\mathrm{Sign}_{sk}(.)$. يخرج الخصم $(m, \sigma)$.
3. ليكن $Q$ يمثل عدد الرسائل التي يوقّعها الخصم $\mathcal{A}$. يكون اخراج التجربة 1 اذا كان $m \notin Q$ و$\mathrm{Vrfy}_{pk}(m, \sigma) = 1$

**تعريف(7.2):** يكون منهج التوقيع $\pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Vrfy})$ **آمنا ضد مهاجمات التزوير من نوع** chosen-message attack (CMA) اذا كان لجميع الخصوم $\mathcal{A}$ ، توجد دالة ضئيلة جدا $\mathrm{negl}$ بحيث ان:

$$\Pr[\mathrm{Sig - forge}^{\mathrm{cma}}_{\mathcal{A},\pi}(n) = 1] \leq \mathrm{negl}(n)$$

## 7.4 تصميم التوقيع الرقمي

في هذا الجزء سنتطرق لكيفية تصميم مناهج التوقيع الرقمي.

### 7.4.1 توقيع RSA



سنقدّم طرق مختلفة لتصميم مناهج التوقيع الرقمي باستخدام RSA.

- نبتدأ بعرض منهج توقيع RSA غير أمن ومن ثم نقدّم نموذج جديد يتماشى مع تعريف التوقيع الرقمي الآمن.

**التوقيع باستخدام RSA غير الآمن.**

لتكن $\mathrm{GenRSA}$ خوارزمية تستلم معامل الأمنية $\mathrm{n}$ وتخرج $\mathrm{N}$ بشكل حاصل ضرب عددي اوليين بطول $\mathrm{n}$ ثنائية ، مع عددين صحيحين $\mathrm{e}, \mathrm{d}$ بحيث $\mathrm{ed} = 1 \bmod \emptyset(\mathrm{n})$. يوضّح المنهج (7.1) طريقة التوقيع الرقمي غير الآمن باستخدام RSA.

يمكن بسهولة ان نرى ان فحص التوقيع الصحيح دائما يكون ممكنا وذلك لان

$$\sigma^{\mathrm{e}} = \left(\mathrm{m}^{\mathrm{d}}\right)^{\mathrm{e}} = \mathrm{m} \bmod \mathrm{N}$$

**Construction (7.1): Insecure RSA–based signature**

1. **Gen**: takes $\mathrm{n}$, runs $\mathrm{GenRSA(n)}$to obtain $(\mathrm{N}, \mathrm{e}, \mathrm{d})$ the public key is $\mathrm{pk} =< \mathrm{N}, \mathrm{e} >$and the private key is $\mathrm{sk} =< \mathrm{N}, \mathrm{d} >$.
2. **Sign**: on input $\mathrm{sk}$ and a message $\mathrm{m} \in \mathbb{Z}_{\mathrm{N}}^{*}$, outputs the signature
$$\sigma = [\mathrm{m}^{\mathrm{d}} \bmod \mathrm{N}].$$
3. **Vrfy**: on input $\mathrm{pk}$, a message $\mathrm{m} \in \mathbb{Z}_{\mathrm{N}}^{*}$, and the signature $\sigma \in \mathbb{Z}_{\mathrm{N}}^{*}$, output 1 if and only if
$$\mathrm{m} = [\sigma^{\mathrm{e}} \bmod \mathrm{N}].$$

- هذا التصميم غير أمن لأن الخصم يستطيع بسهولة اخراج توقيع مزوّر اعتماد على المفتاح المعلن فقط.
- يعمل الخصم كما يلي: يختار $\sigma \in \mathbb{Z}_{\mathrm{N}}^{*}$ ويحسب $\mathrm{m} = [\sigma^{\mathrm{e}} \bmod \mathrm{N}]$ ، ومن ثم يخرج $(\mathrm{m}, \sigma)$.
- لاحظ ، ان فحص التوقيع سوف ينجح ، وهذا يعتبر توقيع مزوّر لكون $\sigma$ لم يوقّع من قبل صاحب المفتاح المعلن وبذلك لايتوافق هذا التصميم مع تعريف التوقيع الرقمي (7.2).

**التوقيع الرقمي ودوال النحت.**

- يتم استخدام دوال النحت (تطرقنا لها في فصل 4) مع التوقيع الرقمي للتخلص من طريقة التزوير السابقة.
- تتلخص الفكرة الاساسية لتحديث التوقيع بطريقة RSA بتطبيق دالة $\mathrm{H}$ على الرسالة قبل توقيعها. لتكن لدينا الدالة $\mathrm{H}: \{0,1\}^{*} \rightarrow \mathbb{Z}_{\mathrm{N}}^{*}$ ، توقّع الرسالة $\mathrm{m} \in \{0,1\}^{*}$ وذلك بحساب

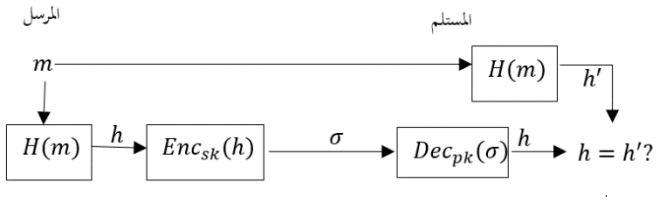
$$\sigma = [\mathrm{H}(\mathrm{m})^{\mathrm{d}} \bmod \mathrm{N}]$$

اما فحص الزوج $(m, \sigma)$ فيتم باختبار

$$\sigma^{e} = H(m) \bmod N.$$



يوضّح الشكل (7.1) كيفية التوقيع باستخدام دوال النحت.




**شكل** (7.1): التوقيع باستخدام دوال النحت وطريقة RSA

- لاحظ ، ان اقل متطلب بالنسبة للدالة $H$ هو ان تكون دالة نحت **مقاومة للتصادم** ، اذا لم تكن كذلك فإن الخصم يستطيع اخراج رسالتين مختلفتين $m_1$ و $m_2$ بحيث يكون لهما $H(m_1) = H(m_2)$ ومن ثم يكون التزوير بسيطا.
- يدعى هذا النوع من التوقيع بـ(hash-and-sign)،

نستطيع ان نصف الأن كيف ان المهاجمة السابقة على منهج RSA للتوقيع اصبحت اكثر صعوبة عند استخدام توقيع RSA المنحوت(hashed-RSA signature). الطريقة الاعتيادية للمهاجمة تصبح بالشكل التالي: يختار الخصم $\sigma \in \mathbb{Z}_N^*$ ويحسب $\hat{m} = [\sigma^e \bmod N]$، ومن ثم يحاول ايجاد رسالة $m \in \{0,1\}^*$ بحيث يكون لها $H(m) = \hat{m}$. عندما تكون الدالة $H$ غير قابلة للعكس بسهولة تكون هذه العملية صعبة.

- بالاضافة للحماية التي يوفرها منهج توقيع RSA المنحوت ضد بعض المهاجمات ، يمتلك هذا المنهج ايضا ميزة اخرى وهي امكانية توقيع رسائل مختلفة الطول $m \in \{0,1\}^*$ بدلا من اشتراط كونها عناصر في الزمرة $\mathbb{Z}_N^*$.
- تجدر الاشارة الى انه لا توجد دالة $H$ يكون فيهل توقيع hashed-RSA آمنا وفق تعريف التوقيع الرقمي (7.2).
- لتقديم منهج توقيع أمن من نوع hashed-RSA فإن الدالة $H$ لانعتبرها دالة نحت مقاومة للتصادم collision-resistance hash function بل تعتبر دالة من نوع **التنبؤ العشوائي** random oracle.

**دوال نحت التنبؤ العشوائي** Random oracle hash functions.

تكون هذه الدوال دوال **نحت مثالية** ideal ، بحيث لايمكن التنبؤ بمخرجاتها $H(x)$ للادخال $x$ الّا عن طريق استعلام تلك الدالة $H(x)$.

- يمكن اعتبار دوال التنبؤ على انها صندوق توضع بداخله تلك الدالة ، بحيث لايسمح لأحد بمعرفة كيفية عمل تلك الدالة.
- كل ماهو ممكن هو فقط استعلام تلك الدالة.
- تقوم الدالة بالاستجابة بصورة عشوائية للادخال الجديد. عند تكرار المدخلات تتكرر المخرجات.

- وبالتالي فإن قيمة $H(x)$ تكون عشوائية تماما ما لم يتم تقديم الادخال $x$ من قبل.
- على الرغم من عدم توفر دوال عشوائية في الواقع ، فاننا نأمل ان تتصرف دوال النحت المصممة جيدا كدوال تنبؤ عشوائي.
- من الناحية العملية يمكن اعتبار دوال النحت مثل SHA-1 ، و SHA3 كتطبيق عملي لدوال التنبؤ العشوائي.



**توقيع RSA في نموذج التنبؤ العشوائي**. بعد ان تم التعرّف بصورة مجملة عن معنى دوال التنبؤ العشوائية ، نعرض الأن تصميم توقيع RSA الآمن. يوضّح المنهج (7.2) هذا التصميم.

- عندما تكون الدالة $H$ هي دالة تنبؤ عشوائي فإن المنهج (7.2)يحقق تعريف التواقيع الرقمية الامنة(7.2).
  - لتوضيح ذلك نفترض بداية ان الخصم يصل فقط الى دالة $H$ بدون الوصول الى خوارزمية التوقيع Sign. عندما يخرج الخصم التزوير $(m, \sigma)$ فاننا نفترض بأنه قام مسبقا بحساب $H(m)$. ليكن $y_1, \dots, y_q$ هي الاجابات التي حصل عليها من $q$ من استعلامات للدالة $H$. لاحظ ، ان $y_i$ هو عشوائي تماما وان تزوير توقيع صالح يتطلب من الخصم حساب $\sigma = y^{1/e} \bmod N$ وهي عملية صعبة ، حسب افتراض RSA. لنفترض الأن ان الخصم يستطيع الوصول الى خوارزمية التوقيع فإن عملية التزوير تصبح اصعب ، لان هذا يتطلب ايجاد توقيع $\sigma$ للرسالة $m$ وبشرط ان يكون $H(m) = [\sigma^e \bmod N]$.

**Construction (7.2): RSA–secure signature scheme**

**Assumption**: Let $H: \{0,1\}^* \to \{0,1\}^{2n}$ be random oracle function.

**Key generation**: Run $\mathrm{GenRSA}(n)$ to obtain $(N, e, d)$. The public key is $pk =< N, e >$ and the private key is $sk =< N, d >$.

**Signing**: To sign a message $m \in \{0,1\}^*$ on input $sk$, computes the signature

$$\sigma = [H(M)^d \bmod N].$$

**Verification**: Given the signature $\sigma$ on a message $m$, with respect to $pk$, output 1 if and only if

$$\sigma^e = [H(m) \bmod N].$$

## 7.4.2 منهج التوقيع بطريقة El Gamal

تعتبر طريقة El Gamal للتوقيع الرقمية طريقة غير محددة non-deterministic ، على العكس من طريقة توقيع RSA ، حيث يكون هناك تواقيع مختلفة عند توقيع الرسالة عدة مرات. تعتمد أمنية هذه الطريقة على صعوة مسألة اللوغاريتم المتقطع discrete logarithm problem. يوضّح المنهج (7.3) كيفية عمل هذه الطريقة.

**Construction (7.3): El Gamal signature scheme**

**Key generation**: Let $p$ be prime number, $g \in \mathbb{Z}_p^*$ be the generator of $\mathbb{Z}_p^*$ group, $q = p - 1$ be the order of $\mathbb{G}$, let $a \in \mathbb{Z}_p^*$ be a private number

- The public key is $pk =< p, q, g, b = g^a \bmod p >$
- The private key is $sk = a$.

**Signing**: To sign the message $m \in \mathbb{Z}_p^*$, with respect to $sk$, first choose a secret random number $k \in \mathbb{Z}_q^*$, and compute the signature

$$\sigma = (\gamma, \delta),$$

$$\text{where } \gamma = [g^k \bmod p], \delta = [(m - a\gamma)k^{-1} \bmod (p-1)].$$

**Verification**: The verification for message $m \in \mathbb{Z}_p^*$ and signature $\sigma = (\gamma, \delta)$ output 1 if and only if

$$b^{\gamma}\gamma^{\delta} = g^m \bmod p.$$



ويمكن التوصل لهذه النتيجة عندما نبدأ بمعادلة الفحص واشتقاق معادلة التوقيع. افترض اننا بدأنا من

$$g^m = b^{\gamma}\gamma^{\delta} \bmod p$$

نقوم الأن بتعويض $\gamma = [g^k \bmod p]$ و $b = g^a$ لنحصل على

$$g^m = g^{a\gamma + k\delta} \bmod p$$

وبما ان $g$ هو مولّد للزمرة في حالة $\bmod p$ ، فإن التطابق يكون صحيحا اذا كانت الاسس متطابقة في حالة $\bmod q$ ، وبالتالي:

$$m = a\gamma + k\delta \bmod q$$

من خلال هذه المعادلة نحصل على قيمة $\delta = [(m - a\gamma)k^{-1} \bmod (p-1)$ المستخدمة في التوقيع.

**مثال(7.1):** ليكن $p = 467, g = 2, a = 127$ عندها

$$b = g^a \bmod p = 2^{127} \bmod 467 = 132.$$

افترض ان Alice ترغب بتوقيع الرسالة $m = 10$ ، وانها اختارت الرقم العشوائي

$$k = 213 \; (k^{-1} \bmod p = 213^{-1} \bmod 466 = 431).$$

يحسب التوقيع كما يلي:

$$\gamma = 2^{213} \bmod 467 = 29$$

$$\delta = (100 - 127 \times 29)431 \bmod 466 = 51$$



يستطيع اي شخص ان يثبت صحة التوقيع بأختبار:

$132^{29}29^{51} = 189 \bmod 467$ و $2^{100} = 189 \bmod 467$ ، وبذلك فإن التوقيع صالح.

**أمنية توقيع El Gamal.**

- افترض ان الخصم Eve تحاول تزوير توقيع للرسالة $m$ ، بدون معرفة المفتاح الخاص $a$.
- اذا استطاعت Eve ان تختار $\gamma$ وتحاول معرفة قيمة $\delta$ المقابلة لها، فانها يجب ان تحسب مسألة اللوغاريتم المتقطع $\log_{\gamma} g^{m} b^{-\gamma}$.
- اما اذا اختارات قيمة $\delta$ اولا ومن ثم تحاول ان تجد $\gamma$ ، فأنها تحاول ان تحل المعادلة

$$b^{\gamma}\gamma^{\delta} \bmod p = g^{m} \bmod p$$

للمجهول $\gamma$. وهذه مسألة لايوجد لها حل متيّسر كما انها غير مرتبطة بمسائل اللوغاريتم المتقطع.

### 7.4.3 منهج توقيع Schnorr

- في العديد من الحالات ، يتم تشفير الرسالة وفك شفرتها مرة واحدة فقط ، لذا يكفي استخدام اي منهج تشفير لهذا الغرض.
- على العكس من ذلك ، تتم عملية اختبار صلاحية التوقيع الرقمي عدد كبير من المرات.
  - لذا من الضروري الاهتمام بكفاءة مناهج التوقيع الرقمي اكثر من مناهج التشفير.
- تحتاج طريقة El Gamal السابقة الى استخدام قيمة $p$كبيرة لكي تصعب حل مسألة اللوغاريتم المتقطع التي استندت عليها.
- العديد من الباحثين يقترح ان لا يقل طول $p$ عن 2048 ثنائية لتوفير أمنية عالية.
- يتطلب استخدام 2048 ثنائية لطول مخرجات طريقة El Gamal ان يكون طول التوقيع 4096 ثنائية.
- بالنسبة للتطبيقات المحتملة التي تستدعي استخدام بطاقات ذكية ، يكون من الافضل استخدام تواقيع اقصر.
- قدّم Schnorr منهج توقيع يعتمد على طريقة El Gamal ولكن بمفتاح اقصر بكثير. نعرض فيما يلي هذا المنهج.

ليكن لدينا عددان اوليّان $p, q$ ، بحيث $p - 1 = 0 \bmod q$. عادة ماتكون $p \approx 2^{2048}$ و $q \approx 2^{224}$ . يكون العمل في طريقة Schnorr ضمن الزمرة $\mathbb{Z}_p^*$ والتي عدد عناصرها $q$. تعتمد طريقة Schnorr على استخدام دالة نحت $H: \{0,1\}^* \rightarrow \mathbb{Z}_q$ آمنة. من المهم ملاحظة ان دالة النحت تعتبر في طريقة Schnorr من ضمن خوارزمية التوقيع ، عكس اسلوب hash-and-sign السابق ، الذي يطبّق دالة النحت على الرسالة ثم يوقع الناتج. يوضّح المنهج (7.4) طريقة عمل هذا التوقيع.



**Construction** (7.4): **Shnorr signature scheme**

**Key generation**: Let $p$ be a prime number, let $q$ be a prime such that divides $p - 1$.

Let $g \in \mathbb{Z}_p^*$ be $q$th root of $1 \bmod p$, let $a \in \{0, \dots, q - 1\}$ be a private number.

The public key is $pk =< p, q, g, b = g^a \bmod p >$.

The private key is $sk = a$. Finally, let $H: \{0,1\}^* \rightarrow \mathbb{Z}_q$ be a secure hash function.

**Signing**: To sign the message $m \in \{0,1\}^*$ with respect to $sk$, first choose a secret random number $k \in \{1, \dots, q - 1\}$ and compute the signature

$$\sigma = (\gamma, \delta), \text{where } \gamma = H(m||g^k \bmod p),$$

$$\delta = [(k + a\gamma) \bmod q].$$

**Verification**: The verification for message $m \in \{0,1\}^*$ and signature $\sigma = (\gamma, \delta)$ output 1 if and only if

$$H(m||b^{-\gamma}g^{\delta} \bmod p) = \gamma.$$

يمكن اثبات صحة عمل خوارزمية الاختبار كما يلي:

$$H(m||b^{-\gamma}g^{\delta} \bmod p) = H(m||g^k \bmod p)$$

نحتاج الى اثبات ان

$$b^{-\gamma}g^{\delta} \bmod p = g^k \bmod p$$

بتعويض $b = g^a \bmod p$ ، $\delta = [(k + a\gamma) \bmod q$ نحصل على

$$g^{-a\gamma+k+a\gamma} \bmod p = g^k \bmod p$$

**مثال(7.2):** ليكن $q = 101, p = 78q + 1 = 7879, g = 170$. افترض ان $a = 170$ بالتالي فإن

$$b = g^a \bmod 7879 = 4567$$

افترض الأن ان Alice ترغب بتوقيع الرسالة $m$ ، لذا سوف تختار رقم عشوائي $k = 50$ ، ثم تحسب

$$g^k \bmod p = 170^{50} \bmod 7879 = 2518$$

تقوم بعدها بحساب $H(m||2518)$ ، حيث $H$ هي دالة نحت. افترض ان $H(m||2518) = 96$. عندها يحسب $\delta$ كما يلي:

$$\delta = 50 + 75 \times 96 \bmod 101 = 79$$

وبالتالي يكون التوقيع هو : $\sigma = (96,79)$.



يفحص هذا التوقيع بحساب: $170^{79}4567^{-96} \bmod 7879 = 2518$ واختبار كون $H(m||2518) = 96$.

### 7.4.4 خوارزمية التوقيع الرقمي

- تعتبر خوارزمية التوقيع الرقمي Digital signature algorithm (DSA) احد التعديلات على طريقة توقيع El Gamal مع ضم بعض الافكار من طريقة Schnorr.
- تم اعتبارها كمعيار للتوقيع الرقمي عام 1994.
- تستخدم DSA الرتبة $q$ للزمرة $\mathbb{Z}_p^*$ كما هو حال طريقة Schnorr.
- يشترط في DSA حاليا ان يكون طول $q$ هو 224 ثنائية و طول $p$ هو 2048 ثنائية.
- توليد مفتاح DSA له نفس شكل طريقة Schnorr.
- نفترض ان الرسالة تنحت قبل توقيعها ، حيث نستخدم دالة النحت SHA3-224 لنحت الرسالة بطول 224 ثنائية. يكون طول الوقيع في DSA هو 448 ثنائية.
- يوضّح المنهج (7.5) طريقة عمل توقيع DSA.

المثال التالي يوضّح كيفية التوقيع باستخدام DSA باستخدام اعداد اولية صغيرة.

**مثال(7.3):** افترض نفس مدخلات المثال السابق. افترض ان Alice ترغب بتوقيع الرسالة $SHA3 - 224(m) = 22$. ثم تحسب

$$k^{-1} \bmod 101 = 50^{-1} \bmod 101 = 99,$$

$$\gamma = (170^{50} \bmod 7879) \bmod 101 = 2518 \bmod 101 = 94,$$

و

$$\delta = (22 + 75 \times 94)99 \bmod 101 = 97.$$

يمكن ان يفحص التوقيع $\sigma = (94,97)$ على الرسالة التي بصمتها 22 بالحسابات التالية:

$$\delta^{-1} = 97^{-1} \bmod 101 = 25$$

$$e1 = 22 \times 25 \bmod 101 = 45$$

$$e2 = 94 \times 25 \bmod 101 = 27$$

$$(170^{45}4567^{27} \bmod 7879) \bmod 101 = 2518 \bmod 101 = 94$$

**Construction (7.5): Digital Signature Algorithm (DSA)**

**Key generation**: Let $p$ be 2048-bit prime number, let $q$ be 224-bit prime such that divides $p-1$. Let $g \in \mathbb{Z}_p^*$ be $q$th root of $1 \bmod p$, let $a \in \{0,\dots,q-1\}$ be a private number. The public key is $pk =< p, q, g, b = g^a \bmod p >$. The private key is $sk = a$. Finally, let $H: \{0,1\}^* \to \mathbb{Z}_q$ be a secure hash function.

**Signing**: To sign the message $m \in \{0,1\}^*$ with respect to $sk$, first choose a secret random number $k \in \{1,\dots,q-1\}$ and compute the signature

$$\sigma = (\gamma, \delta), \text{where } \gamma = (g^k \bmod p) \bmod q,$$

$$\delta = [SHA3-224(m) + a\gamma]k^{-1} \bmod q].$$

(if $\gamma = 0$ or $\delta = 0$, then try another $k$).

**Verification**. The verification for message $m \in \{0,1\}^*$ and signature $\sigma = (\gamma, \delta)$ is done by computing:

$$e1 = SHA3-224(m)\delta^{-1} \bmod q$$

$$e2 = \gamma\delta^{-1} \bmod q$$

and output 1 if and only if

$$(g^{e1}b^{e2} \bmod p) \bmod q = \gamma.$$