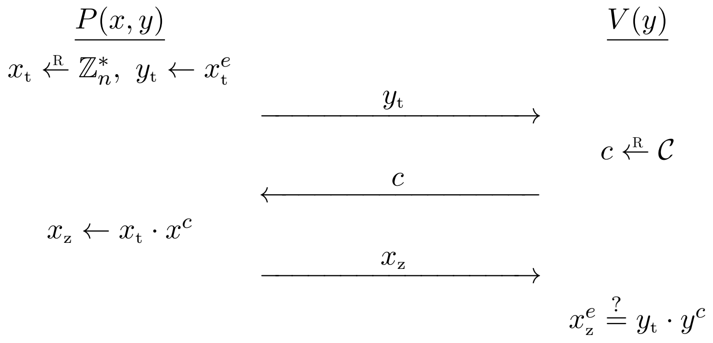
$P(x,y)$
$V(y)$
$x_{\mathrm{t}} \xleftarrow{\mathrm{R}} \mathbb{Z}_n^*,\ y_{\mathrm{t}} \leftarrow x_{\mathrm{t}}^e$
y_{t}
$c \xleftarrow{\mathrm{R}} \mathcal{C}$
c
$x_{\mathrm{z}} \leftarrow x_{\mathrm{t}} \cdot x^c$
x_{z}
$x_{\mathrm{z}}^e \overset{?}{=} y_{\mathrm{t}} \cdot y^c$

Figure 19.10: The GQ protocol

Special soundness. Next, we show that the GQ protocol provides special soundness. Suppose we have two accepting conversations $(x_{\mathrm{t}}, c, x_{\mathrm{z}})$ and $(x_{\mathrm{t}}, c', x'_{\mathrm{z}})$ for the statement y, where $c \neq c'$. We have to show how to efficiently compute an eth root of y. Observe that

$$x_{\mathrm{z}}^e = y_{\mathrm{t}} \cdot y^c \quad \text{and} \quad (x'_{\mathrm{z}})^e = y_{\mathrm{t}} \cdot y^{c'}.$$

Dividing the first equation by the second, we obtain

$$(\Delta x)^e = y^{\Delta c}, \quad \text{where } \Delta x := x_{\mathrm{z}}/x'_{\mathrm{z}},\ \Delta c := c - c'.$$

Observe that because $c \neq c'$ and both c and c' belong to the interval $\{0, \ldots, e-1\}$, we have $0 < |\Delta c| < e$, and so $e \nmid \Delta c$; moreover, since e is prime, it follows that $\gcd(e, \Delta c) = 1$. Thus, we may apply Theorem 10.6 (with the given e, $f := \Delta c$, and $w := \Delta x$), to obtain an eth root of y.

The reader should observe that the technique presented here for computing an RSA inverse from two accepting conversations is essentially the same idea that was used in the proof of Theorem 10.7. Indeed, the two accepting conversations yield a collision $((x_{\mathrm{z}}, -c \bmod e), (x'_{\mathrm{z}}, -c' \bmod e))$ on the hash function $H_{\mathrm{rsa}}(a, b) := a^e y^b$.

Special HVZK. Finally, we show that the GQ protocol is special HVZK by exhibiting a simulator. On input $y \in \mathbb{Z}_n^*$ and $c \in \mathcal{C}$, the simulator computes

$$x_{\mathrm{z}} \xleftarrow{\mathrm{R}} \mathbb{Z}_n^*, \ y_{\mathrm{t}} \leftarrow x_{\mathrm{z}}^e / y^c$$

and outputs $(y_{\mathrm{t}}, x_{\mathrm{z}})$. The key observation is that in a real conversation, c and x_{z} are independent, with c uniformly distributed over $\mathcal{C}$ and x_{z} uniformly distributed over $\mathbb{Z}_n^*$; moreover, given c and x_{z}, the value y_{t} is uniquely determined by the equation $x_{\mathrm{z}}^e = y_{\mathrm{t}} \cdot y^c$. It should be clear that this is the same as the output distribution of the simulator. □

19.6 Identification and signatures from Sigma protocols

By mimicking the Schnorr constructions, we can easily convert any Sigma protocol into a corresponding identification scheme and signature scheme.

Suppose we have a Sigma protocol (P, V) for a relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$. In addition to P and V, we need a **key generation algorithm for** $\mathcal{R}$. This is a probabilistic algorithm G that generates a public-key/secret-key pair (pk, sk), where $pk = y$ and $sk = (x, y)$ for some $(x, y) \in \mathcal{R}$.

To get secure identification and signature schemes we need the following "one-wayness" property: given a public key $pk = y \in \mathcal{Y}$ output by G, it should be hard to compute $\hat{x} \in \mathcal{X}$ such that $(\hat{x}, y) \in \mathcal{R}$. This notion is made precise by the following attack game.

Attack Game 19.2 (One-way key generation). Let G be a key generation algorithm for $R \subseteq \mathcal{X} \times \mathcal{Y}$. For a given adversary $\mathcal{A}$, the attack game runs as follows:

- The challenger runs $(pk, sk) \xleftarrow{R} G()$, and sends $pk = y$ to $\mathcal{A}$;
- $\mathcal{A}$ outputs $\hat{x} \in \mathcal{X}$.

We say that the adversary wins the game if $(\hat{x}, y) \in \mathcal{R}$. We define $\mathcal{A}$'s advantage with respect to G, denoted $\mathrm{OW}\mathsf{adv}[\mathcal{A}, G]$, as the probability that $\mathcal{A}$ wins the game. □

Definition 19.6. *We say that a key generation algorithm G is* ***one way*** *if for all efficient adversaries $\mathcal{A}$, the quantity $\mathrm{OW}\mathsf{adv}[\mathcal{A}, G]$ is negligible.*

Example 19.4. For the Schnorr Sigma protocol (Example 19.1), the most natural key generation algorithm computes $\alpha \xleftarrow{R} \mathbb{Z}_q$ and $u \leftarrow g^\alpha \in \mathbb{G}$, and outputs $pk := u$ and $sk := (\alpha, u)$. It is clear that this key generation algorithm is one-way under the DL assumption. □

Example 19.5. Consider the GQ protocol in Section 19.5.5. Recall that the RSA public key (n, e) is viewed here as a system parameter. The most natural key generation algorithm computes $x \xleftarrow{R} \mathbb{Z}_n^*$ and $y \leftarrow x^e \in \mathbb{Z}_n^*$. It outputs $pk := y$ and $sk := (x, y)$. It is clear that this key generation algorithm is one-way under the RSA assumption (see Theorem 10.5). □

A Sigma protocol (P, V) with a key generation algorithm G gives an identification scheme (G, P, V). The next two theorems prove that it is secure against eavesdropping attacks.

Theorem 19.14. *Let (P, V) be a Sigma protocol for an effective relation $\mathcal{R}$ with a large challenge space. Let G be a key generation algorithm for $\mathcal{R}$. If (P, V) provides special soundness and G is one-way, then the identification scheme $\mathcal{I} := (G, P, V)$ is secure against direct attacks.*

> *In particular, suppose $\mathcal{A}$ is an efficient impersonation adversary attacking $\mathcal{I}$ via a direct attack as in Attack Game 18.1, with advantage $\epsilon := \mathrm{ID1}\mathsf{adv}[\mathcal{A}, \mathcal{I}]$. Then there exists an efficient adversary $\mathcal{B}$ attacking G as in Attack Game 19.2 (whose running time is about* twice *that of $\mathcal{A}$), with advantage $\epsilon' := \mathrm{OW}\mathsf{adv}[\mathcal{B}, G]$, such that*
>
> $$\epsilon' \geq \epsilon^2 - \epsilon/N, \tag{19.17}$$
>
> *where N is the size of the challenge space, which implies*
>
> $$\epsilon \leq \frac{1}{N} + \sqrt{\epsilon'}. \tag{19.18}$$

Proof. We can just mimic the proof of Theorem 19.1. Using the impersonation adversary $\mathcal{A}$, we build an adversary $\mathcal{B}$ that breaks the one-wayness of G, as follows. Adversary $\mathcal{B}$ is given a public key $pk = y$ from its challenger, and our goal is to make $\mathcal{B}$ compute $\hat{x}$ such that $(\hat{x}, y) \in \mathcal{R}$, with help from $\mathcal{A}$. The computation of $\mathcal{B}$ consists of two stages.

In the first stage of its computation, $\mathcal{B}$ plays the role of challenger to $\mathcal{A}$, giving $\mathcal{A}$ the value $pk = y$ as the verification key. Using the same rewinding argument as in the proof of Theorem 19.1, with probability at least $\epsilon^2 - \epsilon/N$, adversary $\mathcal{B}$ obtains two accepting conversations (t, c, z) and (t, c', z') for y with $c \neq c'$. In more detail, $\mathcal{B}$ awaits $\mathcal{A}$'s commitment t, gives $\mathcal{A}$ a random challenge c, and awaits $\mathcal{A}$'s response z. After this happens, $\mathcal{B}$ rewinds $\mathcal{A}$'s internal state back to the point just after which it generated t, gives $\mathcal{A}$ another random challenge c', and awaits $\mathcal{A}$'s response z'. By the Rewinding Lemma (Lemma 19.2), this procedure will yield the two required accepting conversations with probability at least $\epsilon^2 - \epsilon/N$.

In the second stage of the computation, $\mathcal{B}$ feeds these two conversations into a witness extractor (which is guaranteed by the special soundness property) to extract a witness $\hat{x}$ for y.

That proves (19.17), and (19.18) follows by the same calculation as in Theorem 19.1. □

Theorem 19.3 obviously applies to identification protocols derived from special HVZK Sigma protocols:

Theorem 19.15. *Let (P, V) be a Sigma protocol for an effective relation $\mathcal{R}$. Let G be a key generation algorithm for $\mathcal{R}$. If the identification protocol $\mathcal{I} = (G, P, V)$ is secure against direct attacks, and (P, V) is special HVZK, then $\mathcal{I}$ is also secure against eavesdropping attacks.*

> *In particular, for every impersonation adversary $\mathcal{A}$ that attacks $\mathcal{I}$ via an eavesdropping attack, as in Attack Game 18.2, there is an adversary $\mathcal{B}$ that attacks $\mathcal{I}$ via a direct attack on, as in Attack Game 18.1, where $\mathcal{B}$ is an elementary wrapper around $\mathcal{A}$, such that*
>
> $$\mathrm{ID2adv}[\mathcal{A}, \mathcal{I}] = \mathrm{ID1adv}[\mathcal{B}, \mathcal{I}].$$

Example 19.6. If we augment the GQ protocol (P, V) with the key generation algorithm G in Example 19.5, then we get an identification scheme $\mathcal{I}_{\mathrm{GQ}} = (G, P, V)$ that is secure against eavesdropping attacks under the RSA assumption (provided the challenge space is large). □

19.6.1 The Fiat-Shamir heuristic for signatures

We can convert Sigma protocols to signature schemes, using the same technique developed in Section 19.2. The general technique is originally due to Fiat and Shamir. The building blocks are as follows:

- a Sigma protocol (P, V) for a relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$; we assume that conversations are of the form (t, c, z), where $t \in \mathcal{T}$, $c \in \mathcal{C}$, and $z \in \mathcal{Z}$;

- a key generation algorithm G for $\mathcal{R}$;

- a hash function $H : \mathcal{M} \times \mathcal{T} \to \mathcal{C}$, which will be modeled as a random oracle; the set $\mathcal{M}$ will be the message space of the signature scheme.

The **Fiat-Shamir signature scheme** derived from G and (P, V) works as follows:

- The key generation algorithm is G, so a public key is of the form $pk = y$, where $y \in \mathcal{Y}$, and a secret key is of the form $sk = (x, y) \in \mathcal{R}$.

- To sign a message $m \in \mathcal{M}$ using a secret key $sk = (x, y)$, the signing algorithm runs as follows:

  - it starts the prover $P(x, y)$, obtaining a commitment $t \in \mathcal{T}$;

- it computes a challenge $c \leftarrow H(m,t)$;
- finally, it feeds c to the prover, obtaining a response z, and outputs the signature $\sigma := (t,z) \in \mathcal{T} \times \mathcal{Z}$.

- To verify a signature $\sigma = (t,z) \in \mathcal{T} \times \mathcal{Z}$ on a message $m \in \mathcal{M}$ using a public key $pk = y$, the verification algorithm computes $c \leftarrow H(m,t)$, and checks that (t,c,z) is an accepting conversation for y.

Just as we did for Schnorr, we will show that the Fiat-Shamir signature scheme is secure in the random oracle model if the corresponding identification scheme (G,P,V) is secure against eavesdropping attacks. However, we will need one more technical assumption, which essentially all Sigma protocols of interest satisfy.

Definition 19.7 (Unpredictable commitments). *Let (P,V) be a Sigma protocol for $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$, and suppose that all conversations (t,c,z) lie in $\mathcal{T} \times \mathcal{C} \times \mathcal{Z}$. We say that (P,V) has δ-**unpredictable commitments** if for every $(x,y) \in \mathcal{R}$ and $\hat{t} \in \mathcal{T}$, with probability at most δ, an interaction between $P(x,y)$ and $V(y)$ produces a conversation (t,c,z) with $t = \hat{t}$. We say that (P,V) has **unpredictable commitments** if it is has δ-unpredictable commitments for negligible δ.*

Theorem 19.16. *If H is modeled as a random oracle, the identification scheme $\mathcal{I} = (G,P,V)$ is secure against eavesdropping attacks, and (P,V) has unpredictable commitments, then the Fiat-Shamir signature scheme $\mathcal{S}$ derived from G and (P,V) is secure.*

> *In particular, let $\mathcal{A}$ be an adversary attacking $\mathcal{S}$ as in the random oracle version of Attack Game 13.1. Moreover, assume that $\mathcal{A}$ issues at most $Q_{\rm s}$ signing queries and $Q_{\rm ro}$ random oracle queries, and that (P,V) has δ-unpredictable commitments. Then there exist a $(Q_{\rm ro}+1)$-impersonation adversary $\mathcal{B}$ that attacks $\mathcal{I}$ via an eavesdropping attack as in Attack Game 19.1, where $\mathcal{B}$ is an elementary wrapper around $\mathcal{A}$, such that*
>
> $$\mathrm{SIG^{ro}}\mathsf{adv}[\mathcal{A},\mathcal{S}] \leq Q_{\rm s}(Q_{\rm s}+Q_{\rm ro}+1)\delta + \mathrm{rID2}\mathsf{adv}[\mathcal{B},\mathcal{I},Q_{\rm ro}+1].$$

The proof of this theorem is almost identical to that of Theorem 19.7. We leave the details to the reader.

Putting everything together, suppose that we start with a Sigma protocol (P,V) that is special HVZK and provides special soundness. Further, suppose (P,V) has unpredictable commitments and a large challenge space. Then, if we combine (P,V) with a one-way key generation algorithm G, the Fiat-Shamir signature construction gives us a secure signature scheme (that is, if we model H as a random oracle). The Schnorr signature scheme is a special case of this construction.

Just as we did for Schnorr signatures, we could use Lemma 19.6 to reduce from r-impersonation to 1-impersonation; however, a tighter reduction is possible. Indeed, the proof of Lemma 19.8 goes through, essentially unchanged:

Lemma 19.17. *Let (P,V) be a special HVZK Sigma protocol for a relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ that provides special soundness, let G be a key generation algorithm for $\mathcal{R}$, and consider the resulting identification protocol $\mathcal{I} = (G,P,V)$. Suppose $\mathcal{A}$ is an efficient r-impersonation eavesdropping adversary attacking $\mathcal{I}$, as in Attack Game 19.1, with advantage $\epsilon := \mathrm{rID2}\mathsf{adv}[\mathcal{A},\mathcal{I},r]$. Then there exists an efficient adversary $\mathcal{B}$ attacking G as in Attack Game 19.2 (whose running time is about* twice *that of $\mathcal{A}$), with advantage $\epsilon' := \mathrm{OW}\mathsf{adv}[\mathcal{B},G]$, such that*

$$\epsilon' \geq \epsilon^2/r - \epsilon/N, \tag{19.19}$$

where N is the size of the challenge space, which implies

$$\epsilon \leq \frac{r}{N} + \sqrt{r\epsilon'}. \tag{19.20}$$

Using this, we get the following concrete security bound for Theorem 19.16, assuming (P, V) is special HVZK and provides special soundness:

> *Let $\mathcal{A}$ be an efficient adversary attacking $\mathcal{S}$ as in the random oracle version of Attack Game 13.1. Moreover, assume that $\mathcal{A}$ issues at most Q_{s} signing queries and Q_{ro} random oracle queries. Then there exists an efficient adversary $\mathcal{B}$ attacking G as in Attack Game 19.2 (whose running time is about* twice *that of $\mathcal{A}$), such that*
>
> $$\text{SIG}^{\text{ro}}\mathsf{adv}[\mathcal{A}, \mathcal{S}] \leq Q_{\text{s}}(Q_{\text{s}} + Q_{\text{ro}} + 1)\delta + (Q_{\text{ro}} + 1)/N + \sqrt{(Q_{\text{ro}} + 1)\text{OW}\mathsf{adv}[\mathcal{B}, G]}), \tag{19.21}$$
>
> *where N is the size of the challenge space.*

19.6.1.1 The GQ signature scheme

The Fiat-Shamir signature construction above applied to the GQ Sigma protocol (Section 19.5.5) gives us a new signature scheme based on RSA. The scheme makes use of an RSA public key (n, e) as a system parameter, where the encryption exponent e is a large prime. If desired, this system parameter can be shared by many users. We need a hash function $H : \mathcal{M} \times \mathcal{T} \to \mathcal{C}$, where $\mathcal{T}$ is a set into which all elements of $\mathbb{Z}_n^*$ can be encoded, $\mathcal{M}$ is the message space of the signature scheme, and $\mathcal{C}$ is a subset of $\{0, \dots, e-1\}$. The GQ signature scheme is $\mathcal{S}_{\text{GQ}} = (G, S, V)$, where:

- The key generation algorithm G runs as follows:

  $$x \xleftarrow{\text{R}} \mathbb{Z}_n^*, \quad y \leftarrow x^e.$$

  The public key is $pk := y$, and the secret key is $sk := x$.

- To sign a message $m \in \mathcal{M}$ using a secret key $sk = x$, the signing algorithm runs as follows:

  $$\begin{aligned} S(\ sk,\ m\ ) := \quad & x_{\text{t}} \xleftarrow{\text{R}} \mathbb{Z}_n^*, \quad y_{\text{t}} \leftarrow x_{\text{t}}^e, \quad c \leftarrow H(m, y_{\text{t}}), \quad x_{\text{z}} \leftarrow x_{\text{t}} \cdot x^c \\ & \text{output } \sigma := (y_{\text{t}}, x_{\text{z}}). \end{aligned}$$

- To verify a signature $\sigma = (y_{\text{t}}, x_{\text{z}})$ on a message $m \in \mathcal{M}$, using the public key $pk = y$, the signature verification algorithm V computes $c := H(m, y_{\text{t}})$. It outputs accept if $x_{\text{z}}^e = y_{\text{t}} \cdot y^c$, and outputs reject, otherwise.

As we saw in Example 19.6, the GQ identification scheme is secure against eavesdropping attacks under the RSA assumption (provided the challenge space is large). Also, we observe that the GQ Sigma protocol has $1/\phi(n)$-unpredictable commitments. It follows from Theorem 19.16 that the corresponding signature scheme is secure in the random oracle model, under the RSA assumption.

The advantage of GQ signatures over RSA signatures, such as $\mathcal{S}_{\text{RSA-FDH}}$, is that the signing algorithm is much faster. Signing with $\mathcal{S}_{\text{RSA-FDH}}$ requires a large exponantiation. Signing with GQ requires two exponentiations with exponents e and c, but both can be only 128 bits. Fast signing is important when the signer is a weak device, as in the case of a chip enabled creditcard that signs every creditcard transaction.