8 Bit Commitment Schemes with Polynomial Time Prover

So far, it was really practical to assume that Peggy (**P**), the prover, had an infinite computational power. We will now put ourselves in a different context by restricting the power of **P**. As we will see, this also has its advantages...

So, we now seek the following:

Binding:	Computational
Concealing:	Perfect/Statistical

Here, the game will be quite different since we will no longer assume that the prover has infinite power, but rather that he is only **probabilistic polynomial time** bounded, just as the verifier, *Vic* (**V**), is.

That is, the prover is no more powerful than the verifier.

Therefore, **P**may know the solution of a particular problem, but be unable to write a solution for it from scratch (!)

Example 8.1 : Factoring
Say Peggy picks two large prime numbers p and q and then let $n := p \cdot q$, BUT she can no longer factor a given n' easily if she has no prior knowledge about it.

Example 8.2 : Graph Isomorphism

$$\boxed{\begin{array}{lcl}
\mathbf{P} & G_0 \approx G_1 & \mathbf{V} \\
 & \xleftarrow{ZK(G_0 \approx G_1)} & \\
commit(b) & & \\
G \approx G_b & \xrightarrow{G} & \\
unveil(b) & \xrightarrow{b,\pi} & G = \pi(G_b)
\end{array}}$$

This is perfectly concealing, but here Peggy is unable to find by herself an isomorphism between the graphs.

8.1 Based on Factoring

P		**V**
		Pick large primes p,q.
		Let $n := p \cdot q$.
		Let $z := x^2 \bmod n$.
	$\xleftarrow{n,z}$	
	$\xleftarrow{ZK(z \in QR_n)}$	
$commit(b)$		
Pick $r \in_R \mathbb{Z}_n$	$\xrightarrow{r^2 \cdot z^b \bmod n}$	
$unveil(b)$	$\xrightarrow{r,b}$	

Notes 8.3 :

- You don't need an infinite powerful prover for the QR_n decision problem, so **V** *can* show here in ZK that $z \in QR_n$.
- b is not uniquely defined. Indeed, it is possible that $r_0^2 \cdot z^0 = r_1^2 \cdot z^1$. If so, $r_0^2 = r_1^2 \cdot z$ and then $(r_0/r_1)^2 = z$. That is, r_0/r_1 is a square root of z, which is, as we know, equivalent to factoring.
- For r chosen randomly, the distribution of $r^2 \bmod n$ is the same as the distribution of $r^2 \cdot z \bmod n$. Hence, the quantity $r^2 \cdot z^b \bmod n$ is meaningless to **V**.
- Concealing, here, is only statistical. Indeed, suppose that we have a $z \in QNR_n$. If **P**becomes convinced that $z \in QR_n$ even if the truth is that $z \in QNR_n$ (recall that this could happen with an exponentially small probability), then the above process won't be concealing.

  On the other hand, the concealing will be perfect if we really have $z \in QR_n$.
- **V**can determine easily if some element is or isn't a quadratic residue since he knows the factors p and q !

Now, let b_1 and b_2 be two bits involved in the above process.
Then, $(r^2 \cdot z^{b_1}) \cdot (s^2 \cdot z^{b_2}) = (rs)^2 \cdot z^{b_1+b_2}$. So we have:

If	Then
$b_1 = b_2 = 0$	$\sqrt{(rs)^2 \cdot z^{b_1+b_2}} = rs$
$b_1 = b_2 = 1$	$\sqrt{(rs)^2 \cdot z^{b_1+b_2}} = rs \cdot z$
$b_1 \neq b_2$	$\sqrt{(rs)^2 \cdot z} = rs \cdot \underbrace{\sqrt{z}}_{\text{We don't know it!}}$

Finally, we get:

$$b_1 = b_2 \iff \mathbf{P}\text{knows } \sqrt{(rs)^2 \cdot z^{b_1+b_2}}$$

Furthermore, we can introduce the following modifications in order to be able to commit to several bits at the same time:

P		**V**
		Pick large primes p,q.
		Let $n := p \cdot q$.
		Pick $z_1, \ldots, z_k \in QR_n$.
	$\xleftarrow{n,z_1,\ldots,z_k}$	
	For each i, $\xleftarrow{ZK(z_i \in QR_n)}$	
$commit(b_1 \ldots b_k)$		
Pick $r \in_R \mathbb{Z}_n$	$\xrightarrow{r^2 \cdot z_1^{b_1} \ldots z_k^{b_k} \bmod n}$	
$unveil(b_1 \ldots b_k)$	$\xrightarrow{r,b_1,\ldots,b_k}$	

As above, for r chosen randomly, the quantity $(r^2 \cdot z_1^{b_1} \ldots z_k^{b_k} \bmod n)$ is a random quadratic residue.

8.2 Based on Discrete Log (DLP)

Let p be a (strong) prime, g a generator of $\mathbb{Z}_p^*$ and $\alpha := g^s \bmod p$, for some power s.

P		**V**
commit(b)		
Pick a random r	$\xrightarrow{g^r \cdot \alpha^b \bmod p}$	
unveil(b)	$\xrightarrow{r,b}$	

The task of Vic is to convince Peggy that he *knows* the discrete logarithm of α.

Note 8.4 : In order to test whether p is prime or not, one can use Rabin's test[1]. In this case, it would be statistically concealing. We can make it perfectly concealing by simply requesting that "**P** *is willing to participate*"[2].

For given r_0 and r_1, we have that:

$$g^{r_0} \cdot \alpha^0 = g^{r_1} \cdot \alpha^1 \Longrightarrow g^{r_0} = g^{r_1} \cdot \alpha \Longrightarrow g^{r_0 - r_1} \equiv \alpha \pmod{p}$$

Therefore,

Break the binding property
is equivalent to
Find the discrete logarithm of α

Hence, the protocol is computationally binding.

Finally, one can do a similar generalization as in the preceding case (factoring) by replacing the single value α by, say, $\alpha_1, \alpha_2, \ldots, \alpha_k$.

8.3 Interactive Arguments

Although so far we did not have any trouble going from an infinite power prover to a probabilistic polynomial time one, it is not always the case.

[1] One could also use the probabilistic algorithm of Goldwasser-Killian that works in polynomial time ($\approx n^{32}$).

[2] Meaning that **P** will collaborate (or be forced by an external judge to collaborate!) until the end of the protocol is reached.

Example 8.5 :
For the three colorability of a graph, $3 - COL$, if we would follow the same approach as before, it would *still* be zero-knowledge, but would not be an interactive *proof* anymore. Indeed, an evil *all powerful* **P**' could prove false statements to **V**. However, an evil *polynomial time* **P**' cannot succeed to prove false statements to **V**with high probability (because of the *bit commitment* on the coloring of the graph).

This motivates the following definition:

Definition 8.6 :
Let $\mathcal{L}$ be a language. A *question-answer* protocol satisfying:

- $\forall x \in \mathcal{L}$, Pr[(**P**,**V**)(x) accepts] $> 2/3$
- $\forall x \notin \mathcal{L}$, $\forall$ **P**' *probabilistic polynomial time*, Pr[(**P**',**V**)(x) accepts] $< 1/3$

is called an *interactive argument* or a *computationally sound interactive proof.*

Remark 8.7 : Although the terminology *interactive argument* is much used in the litterature, we will here prefer *computationally sound interactive proof* since in our exposition, we first started with interactive proofs and *then*, weakened the concept.

Note 8.8 : With computationally sound interactive proofs, breaking of the commitment *has* to be done "on line". So, unlike interactive proofs, we cannot come back after the protocol is over and work off line. Hence, if we want to cheat, it has to be done in *real time.*