MTAT.07.003 Cryptology II
Spring 2012 / Exercise session ?? / Example Solution

Exercise (Guillou-Quisquater identification scheme). *The Guillou-Quisquater identification scheme (GQ scheme) is directly based on the RSA problem. The identification scheme is a honest verifier zero-knowledge proof that the prover knows x such that $x^e = y \mod n$ where n is an RSA modulus, i.e., the public information $\mathsf{pk} = (n, e, y)$ and the secret is x. The protocol itself is following.*

$$\begin{tikzcd}
\mathcal{P} & & \mathcal{V} \\
r \xleftarrow[u]{} \mathbb{Z}_n & & \\
 \arrow[rr, "\alpha = r^e"] & & \\
 & & \beta \xleftarrow[u]{} \{0,1\} \\
 & & \arrow[ll, "\beta"'] \\
 \arrow[rr, "\gamma = r x^{\beta}"] & & \\
 & & \left[\gamma^e \overset{?}{=} \alpha y^{\beta}\right]
\end{tikzcd}$$

Prove that the Guillou-Quisquater identification scheme is a sigma protocol and amplify soundness guarantees with parallel and sequential composition and derive the corresponding knowledge bounds.

Solution. As the protocol has the right message structure, we must show only that the protocol is functional and has special soundness and zero-knowledge property.

Functionality. According to the protocol definition, we can deduce

$$\gamma^e = (r x^{\beta})^e = r^e (x^e)^{\beta} = \alpha y^{\beta}$$

when both parties are honest and thus the verifier reaches accepting state.

Special soundness property. We need to show that there exists an efficient algorithm that can extract the secret knowledge given transcripts of two successful protocol runs with same α message. As the challenge space is $\{0,1\}$, the existence of colliding transcripts means that we observe accepting transcripts $(\alpha_0, 0, \gamma_0)$ and $(\alpha_1, 0, \gamma_1)$. If the prover is semihonest then it is easy to see that the ratio

$$\xi = \frac{\gamma_1}{\gamma_0} = \frac{rx}{r} = x$$

is the desired secret. If the prover is malicious, we can still conclude that

$$\xi^e = \left(\frac{\gamma_1}{\gamma_0}\right)^e = \frac{\gamma_1^e}{\gamma_0^e} = \frac{\alpha y}{\alpha} = y \enspace ,$$

as both transcripts are accepting. Hence, the ratio ξ is always the desired secret.

Zero-knowledge property. For zero-knowledge, we must to show that we are able to simulate protocol transcripts between the honest prover and honest verifier. Differently form the real protocol execution, we can generate the messages in different order than they appear in the protocol. Let us first construct a simulator that correctly creates the first and last message if the challenge β is known ahead. For $\beta = 0$, the protocol transcripts are in the form $(\alpha, 0, r)$. If $\beta = 1$ then protocol transcripts are in the form $(\alpha, 1, rx)$ for uniformly distributed α. If x is invertible, then rx is also uniformly distributed over $\mathbb{Z}_n$ and we can use the following simulator construction:

```
$\mathcal{S}(\beta)$
    $\gamma \xleftarrow[u]{} \mathbb{Z}_n$
    $\alpha \leftarrow \gamma^e \cdot y^{-\beta}$
    **return** $(\alpha, \beta, \gamma)$ .
```

For $\beta = 0$, the correspondence between the real and simulated runs are evident

$$\gamma = r \quad \Longrightarrow \quad \alpha = r^e = \gamma^e \ .$$

The acceptance criterion $\gamma^e = \alpha y$ for $\beta = 1$ assures that for fixed β and γ there is only one α value $\gamma^e \cdot y^{-1}$. Hence, the simulator generates the same triples as real protocol run.

If x is not invertible, then the simulation construction described above fails. Indeed, if $x = 0$ then $\gamma = 0$ whenever $\beta = 1$ while the simulator generates all possible elements of $\mathbb{Z}_n$ for γ. Thus we need a more advanced simulator. First, note that rx and $x^e r$ generate the same distribution for $r \xleftarrow[u]{} \mathbb{Z}_n$. The claim is follows directly from the Chinese Reminder Theorem. If $x \equiv 0 \pmod p$ then $x^e \equiv 0 \pmod p$ for a prime factor of n. Hence, xr and $x^e r$ generate both uniform distribution over the set $p\mathbb{Z}_n$. As a result, we can use the following simulator:

```
S(β)
  $r_* \xleftarrow[u]{} \mathbb{Z}_n$
  $\gamma \xleftarrow[u]{} y^\beta \cdot r_*$
  $\alpha \leftarrow \gamma^e \cdot y^{-\beta}$
  return $(\alpha, \beta, \gamma)$ .
```

The details that the simulator indeed works as expected are left for the reader.

Note that we can use the simulator $\mathcal{S}$ together with message guessing to simulate the conversation with honest prover to any verifier by using the following algorithm:

```
Sim(ϕ, y, n)
  $\omega \leftarrow \Omega$
  For $i \in \{1, \dots, \ell\}$ do
    $\beta_* \xleftarrow[u]{} \{0, 1\}$
    $(\alpha, \beta_*, \gamma) \leftarrow \mathcal{S}(\beta_*)$
    $\beta \leftarrow \mathcal{V}_*(\phi, \alpha; \omega)$
    if $\beta = \beta_*$ then return $\mathcal{V}_*(\gamma)$
  return $\perp$
```

where ϕ is the auxiliary input of the malicious verifier $\mathcal{V}_*$ which captures the information gathered by $\mathcal{V}_*$ before the execution of the Guillou-Quisquater identification scheme. By convention, we restart $\mathcal{V}_*$ and seed with the same randomness in each cycle. As α is independent form β_* the probability that $\beta = \beta_*$ is exactly $\frac{1}{2}$ and thus the simulator returns failure $\perp$ with probability $2^{-\ell}$.

It can be shown that the statistical distance between the output distribution of a malicious verifier $\mathcal{V}_*$ and the output distribution of the simulator Sim are exactly 2^{ℓ}, i.e., we can see the difference only if Sim fails. Exact details are let as an exercise to the reader.

Soundness amplification. By guessing the challenge and using the dedicated simulator $\mathcal{S}$ we can clearly bypass the verification with probability $\frac{1}{2}$. Hence, single protocol instance has knowledge error $\kappa = \frac{1}{2}$. By running the protocol sequentially ℓ times the probability that we succeed all these runs is $2^{-\ell}$. By running all these protocol in parallel we get again a sigma protocol with three rounds:

$$\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
r_1, \dots, r_\ell \xleftarrow[u]{} \mathbb{Z}_n & & \\
 & \xrightarrow{\alpha_1 = r_1^e, \dots, \alpha_\ell = r_\ell^e} & \\
 & & \beta_1, \dots, \beta_\ell \xleftarrow[u]{} \{0,1\} \\
 & \xleftarrow{\beta_1, \dots, \beta_\ell} & \\
 & \xrightarrow{\gamma_1 = r_1 x^{\beta_1}, \dots, \gamma_\ell = r_1 x^{\beta_\ell}} & \\
 & & \bigwedge_{i=1}^{\ell} \left[\gamma_i^e \overset{?}{=} \alpha_i y^{\beta_i}\right]
\end{array}$$

Again, we can show that trivial guessing strategy leads to the success probability $2^{-\ell}$ and thus the knowledge error for this protocol is also $2^{-\ell}$. However, the protocol has several drawbacks if we consider knowledge extraction and simulation for malicious verifier. As the challenge space is exponential in ℓ, the guessing strategy for simulation outlined above is very inefficient for moderately large values of $\ell \geq 80$. Similarly, the challenge space is so big that we cannot look through all potential challenges when we are probing a malicious prover in knowledge extraction. This leads to much looser security guarantees.