16-124-836
Marcel Matti ZAUDER

6.1 Soundness Error

6.1.1 ZKP for Graph Isomorphism

If the Prover $\mathbb{P}$ wants to cheat, the generated graph H would be only isomorphic to either G_0 and G_1 but not both. Therefore in each iteration the Verifier $\mathbb{V}$ can catch P cheating with a probability $\frac{1}{2}$. Therefore the soundness error for the ZKP for Graph Isomorphism with k iterations would be $\frac{1}{2^k}$.

6.1.2 ZKP of knowledge of a discrete logarithm (Schnorr Proof)

If a Prover $\mathbb{P}$ wants to cheat, it needs to correctly guess the value of the challenge before the commitment is made, so it can construct a commitment t, which passes the verification without knowing x s.t. $g^x = y$. This can be done with probability $\frac{1}{q}$. Because this is directly dependant on the security parameter q, multiple verification rounds are not needed if q is sufficiently large.

6.1.3 ZKP of knowledge of an RSA-inverse

With the same argumentation as above the soundness error for the RSA-inverse is $\frac{1}{e}$, because the challenge is chosen from $\mathbb{Z}_e$. Because usually e is often chosen as a rather small parameter, multiple verification rounds might be necessary to lower the soundness error to be of a small enough tolerance.

6.2 Proof-of-knowledge protocol of a representation (REP) [for $n = 2$]

Soundness

We have the two transcripts (t, c, s_1, s_2) and (t, c', s'_1, s'_2), where:

$$t = g_1^{r_1} \cdot g_2^{r_2}$$

Furthermore we have:

Computation of x_1	**Computation of x_2**
$s_1 = r_1 - c \cdot x_1$	$s_2 = r_2 - c \cdot x_2$
$s'_1 = r_1 - c' \cdot x_1$	$s'_2 = r_2 - c' \cdot x_2$
$\Rightarrow s_1 + c \cdot x_1 = s'_1 + c' \cdot x_1$	$\Rightarrow s_2 + c \cdot x_2 = s'_2 + c' \cdot x_2$
$\Leftrightarrow c \cdot x_1 - c' \cdot x_1 = s'_1 - s_1$	$\Leftrightarrow c \cdot x_2 - c' \cdot x_2 = s'_2 - s_2$
$\Leftrightarrow x_1 \cdot (c - c') = s'_1 - s_1$	$\Leftrightarrow x_2 \cdot (c - c') = s'_2 - s_2$
$\Leftrightarrow x_1 = \frac{s'_1 - s_1}{c - c'}$	$\Leftrightarrow x_2 = \frac{s'_2 - s_2}{c - c'}$

Therefore we can both compute x_1 and x_2 and the soundness is shown.

Zero-Knowledge

Verfier $\mathbb{V}$ itself can produce (t, c, s_1, s_2) which satisifes the protocol:

$$\begin{aligned} c &\leftarrow \mathbb{Z}_q \\ s_1, s_2 &\leftarrow \mathbb{Z}_q \\ t &\leftarrow \prod_{i=1}^{2} (g_i^{s_i} \cdot y^c) \end{aligned}$$

6.3 Encrypting a vote

6.3.1 Protocol and ZKPK that allows a party $\mathbb{P}$ to prove that it knows the encrypted value of a value $i \in \mathbb{Z}_q$

Given a value $i \in \mathbb{Z}_q$, the return tuple of the additive ElGamal encryption function would be:

$$(R, C) = AM - \text{ENC}(y, i) = (g^r, g^i \cdot y^r)$$

We want to prove the knowledge of i, s.t. (R, C) is valid encryption of this value ($\mathbb{P}$ knows r, i and $\mathbb{V}$ knows (R, C), additionally both know the public key y):

Prover $\mathbb{P}$		**Verifier** $\mathbb{V}$
$r_1, r_2 \leftarrow \mathbb{Z}_\mathbb{q}$		
$t = g^{r_1} \cdot y^{r_2}$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \leftarrow \mathbb{Z}_q$
$s_1 = r_1 - c \cdot i$		
$s_2 = r_2 - c \cdot r$	$\xrightarrow{s_1, s_2}$	$t \stackrel{?}{=} g^{s_1} \cdot y^{s_2} \cdot C^c$

Because this is a modification of the proof of representation which is given in the lecture, the ZKPK properties obviously hold.

6.3.2 Protocol to encrypt v and prove correctness of encrypted vote to $\mathbb{V}$

The Prover $\mathbb{P}$ wants to prove that $(R, C) = (g^r, g^v \cdot y^r)$ is a valid encryption of $v \in \{0, 1\}$. An equivalent proof is:

$$\log_g(R) = \log_y(\frac{C}{g^0}) \lor \log_g(R) = \log_y(\frac{C}{g^1})$$

Such a statement can be proven by a proof-of-equality (EQ-proof), which we have seen in the lecture. Furthermore to prove that the Prover $\mathbb{P}$ knows either the left or right condition, a proof-of-disjuntion (OR-proof is used). With this we can create the following protocol (here we assume that $v = 1$, for the case $v = 0$, we can just adjust the variables):

Prover $\mathbb{P}$		**Verifier** $\mathbb{V}$
Real proof of $v = 1$		
$\tilde{r} \leftarrow \mathbb{Z}_q$ (blinding factor for EQ)		
$t_1 = g^{\tilde{r}}$		
$t_2 = y^{\tilde{r}}$		
Simulated proof of $v = 0$		
$\hat{c} \leftarrow \mathbb{Z}_q$		
$\hat{s} \leftarrow \mathbb{Z}_q$		
$\hat{t_1} = g^{\hat{s}} \cdot R^{\hat{c}}$		
$\hat{t_2} = y^{\hat{s}} \cdot (\frac{C}{g^0})^{\hat{c}}$		
	$\xrightarrow{t_1, t_2, \hat{t_1}, \hat{t_2}}$	
	$\xleftarrow{\tilde{c}}$	$\tilde{c} \leftarrow \mathbb{Z}_q$
$c = \tilde{c} + \hat{c}$		
$s = \tilde{r} - c \cdot r$	$\xrightarrow{s, c, \hat{c}, \hat{s}}$	$t_1 \stackrel{?}{=} g^s \cdot R^c$ and $t_2 \stackrel{?}{=} y^s \cdot (\frac{C}{g^1})^c$
		$\hat{t_1} \stackrel{?}{=} g^{\hat{s}} \cdot R^{\hat{c}}$ and $\hat{t_2} \stackrel{?}{=} y^{\hat{s}} \cdot (\frac{C}{g^0})^{\hat{c}}$
		$c \stackrel{?}{=} \tilde{c} + \hat{c}$

Again because this protocol is a modification of the proof-of-equality the ZKPK properties hold.