Advanced Topics on Privacy-Enhancing Technologies
CS-523
Anonymous Authentication Exercises - Solutions

1 Zero-knowledge color-blindness

Alice has two pens. They are identical, except that one pen is red and the other blue. Bob is colorblind, so to him the pens look the same. Alice wants to convince Bob that she can distinguish these pens, without revealing to Bob which pen is red and which is blue. To this end, Alice and Bob run the following protocol:

1. (commitment) Alice shuffles the pens and gives them, in a specific order, to Bob. One pen for each hand.
2. (challenge) Bob hides the pens from Alice's view and either (a) swaps the pens, or (b) keeps each pen in the same hand. Each with probability $\frac{1}{2}$. Bob shows the pens to Alice again.
3. (response) Alice tells Bob whether he swaps the pens or not. Bob rejects if Alice's answer is wrong and accepts otherwise.

Prove that this protocol is *complete*, *sound*, and *zero-knowledge*. What is the soundness error?

Solution. Let's argue for completeness. We must show that an honest prover (e.g., a prover that can distinguish the pens) can convince an honest verifier. Since Alice can distinguish the pens, she will always be able to tell whether Bob switched the pens. Therefore, she will always provide the correct answer in step 3. Therefore Bob is always convinced.

To show that the protocol is sound, we must argue that a cheating prover (e.g., a prover that cannot distinguish the pens) cannot convince Bob, except with a small probability. Let's look at a single round. If Alice cannot distinguish the pens, she cannot say whether Bob swaps the pens or not. So Bob will reject with probability $\frac{1}{2}$. We can reduce this error by repeating the protocol n times. The probability that Bob will not reject after n rounds is therefore: $\frac{1}{2^n}$.

Let's prove zero-knowledge. In this case, a transcript consists of a commitment, followed by a challenge and a response. We can easily compute all valid traces:

- commitment: `br`, challenge: `br`, response: same
- commitment: `br`, challenge: `rb`, response: swapped
- commitment: `rb`, challenge: `rb`, response: same
- commitment: `rb`, challenge: `br`, response: swapped

clearly these are indistinguishable from interacting with any verifier.

2 Proving knowledge of a Pedersen commitment

Let $\mathbb{G}$ be a cyclic group of prime order q, generated by g. Let h be another random generator of $\mathbb{G}$.

1. Construct the Sigma protocol for proving knowledge of a Pedersen commitment $\mathsf{com} = g^x h^r$ to the value $x \in \mathbb{Z}_p$. What are the prover's secrets? What are the public values that both the prover and verifier know?
2. Prove completeness, special-soundness and honest-verifier zero-knowledge.
3. Apply the Fiat-Shamir heuristic to your protocol to obtain a non-interactive version.

Solution. The protocol would run as follows:

Common input: group $(\mathbb{G}, g, q)$, h, com		
Prover Input: $x, r \in \mathbb{Z}_q$		**Verifier**
$r_x \in_R \mathbb{Z}_q$		
$r_r \in_R \mathbb{Z}_q$		
$R = g^{r_x} h^{r_r}$	$\xrightarrow{R}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_q$
$s_x = r_x - c \cdot x \pmod q$		
$s_r = r_r - c \cdot r \pmod q$	$\xrightarrow{s_x, s_r}$	Verify $R = \mathsf{com}^c g^{s_x} h^{s_r}$

For completeness we must show that an honest verifier accepts interactions with an honest prover. Since the prover is honest, we know it follows the protocol and that $\mathsf{com} = g^x h^r$. Therefore:

$$\begin{aligned} \mathsf{com}^c g^{s_x} h^{s_r} &= g^{cx} h^{cr} g^{r_x - c \cdot x} h^{r_r - c \cdot} \\ &= g^{r_x} h^{r_r} = R, \end{aligned}$$

and the verifier accepts.

We prove special soundness. Let (R, c, s_x, s_r) and (R, c', s'_x, s'_r) be two accepted traces. Therefore:

$$\mathsf{com}^c g^{s_x} h^{s_r} = \mathsf{com}^{c'} g^{s'_x} h^{s'_r}$$

Reordering, we find that:

$$\mathsf{com} = g^{\frac{s'_x - s_x}{c-c'}} h^{\frac{s'_r - s_r}{c-c'}}$$

Therefore, we recovered a solution $x = \frac{s'_x - s_x}{c-c'}$ and $r = \frac{s'_r - s_r}{c-c'}$.

To prove the zero-knowledge property for honest verifiers, we must simulate traces. In this case, traces take the form (R, c, s_x, s_r). Because we know the verifier is honest, we can assume that c is drawn randomly from $\mathbb{Z}_q$. We can therefore construct a trace as follows. Pick $s_x, s_r, c \in_R \mathbb{Z}_q$ and then compute $R = \mathsf{com}^c g^{s_x} h^{s_r}$. By construction this trace would have been accepted by the verifier. Furthermore, the distributions of the other values are also correct. Notice that in a true execution, r_x, r_r are drawn uniformly at random from $\mathbb{Z}_q$, therefore s_x and s_r are also uniform in $\mathbb{Z}_q$, as they are in our simulation. Finally, c is drawn uniformly by honest verifiers. So this simulated trace is indistinguishble from a real trace.

To apply the Fiat-Shamir heuristic, we compute the challenge c by hashing all the public values as well as the prover's commitment R and an optional message m:

$$c = H(g \parallel h \parallel \mathsf{com} \parallel R \parallel m),$$

where $H : \{0,1\}^* \to \mathbb{Z}_q$ maps strings to $\mathbb{Z}_q$. Traditionally, we do not include a description of the group $\mathbb{G}$.

3 Domain-specific pseudonyms

Consider a credential scheme with a single attribute – the users private key x – that is constructed using blind signatures. In this exercise, a credential then takes the form of a signatures σ on a commitments $C = g^x h^r$ where x is the user's private key.

In this exercise we will work with domain specific pseudonyms to ensure that users will always derive the same pseudonym for the same service provider (but pseudonyms between service providers are unlinkable). The pseudonym nym for a service provider at domain domain is computed as:

$$\mathsf{nym} = H(\mathsf{domain})^x$$

where $H : \{0,1\}^* \to \mathbb{G}$ maps strings to group elements.

Suppose a user wants to use her signature σ to convince a service provider that nym computed as before is her pseudonym. What protocol do the user and the service provider run? If you need to use a zero-knowledge proof, give both the high-level description, and the low level details.

Solution. First, the user computes $\mathsf{nym} = H(\mathsf{domain})^x$ and sends nym, σ and C to the service provider (SP). The verifier checks that σ is a valid signature on C.

Next, the user and the SP engage in the following zero-knowledge proof to convince the SP that the pseudonym nym has been constructed correctly:

$$\mathrm{ZK}\{(x, r) : C = g^x h^r \ \wedge\ \mathsf{nym} = H(\mathsf{domain})^x\}.$$

In more detail, this protocol runs as follows:

Common input: group $(\mathbb{G}, g, q)$, h, domain, C, nym

Prover Input: $x, r \in \mathbb{Z}_q$		**Verifier**
$r_x \in_R \mathbb{Z}_q$ $r_r \in_R \mathbb{Z}_q$ $R_C = g^{r_x} h^{r_r}$ $R_{\mathsf{nym}} = H(\mathsf{domain})^{r_x}$	$\xrightarrow{R_C, R_{\mathsf{nym}}}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_q$
$s_x = r_x - c \cdot x \pmod{q}$ $s_r = r_r - c \cdot r \pmod{q}$	$\xrightarrow{s_x, s_r}$	Verify $R_C = C^c g^{s_x} h^{s_r}$ Verify $R_{\mathsf{nym}} = \mathsf{nym}^c H(\mathsf{domain})^{s_x}$

4 What if verifiers are dishonest

The sigma protocols we constructed in the previous questions assume that the verifier is honest. What goes wrong if the verifier in question 2 is not honest. (Think about how you would construct a trace that cannot be easily simulated.) Could you extend the protocol to make it zero-knowledge even against malicious verifiers?

Solution. All the verifier would have to do is apply the Fiat-Shamir heuristic and compute the challenge based on the prover's commitment. The resulting traces cannot be simulated!

The problem with the Sigma protocol is that the verifier can pick the challenge after the prover has revealed the commitment. However, we can ensure that the verifier picks his challenge *before* seeing the prover's commitment by asking the verifier to commit to its challenge first. Next the prover sends her own commitment. Finally, the verifier reveals the challenge (and shows that it is the challenge that he committed to before). We can prove that this approach is fully zero-knowledge.