ETH Zurich, Department of Computer Science Dr. Jonathan Bootle
HS 2023 Karen Klein, Varun Maram, Antonio Merino Gallardo

Zero-Knowledge Proofs
Exercise 5 (graded)

Submission Deadline: 20/10/2023, 23:59 CEST

Note: Solutions must be typeset in LaTeX. Make sure to name the pdf file of your solutions in the following format:
"*<Last Name>_<First Name>_<Legi Number>_5.pdf*"

5.1 The Closest String Problem (20 marks)

Let $d(\mathbf{u}, \mathbf{v})$ denote the Hamming distance between two binary strings $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_2^m$. Consider the *closest string* relation

$$\mathcal{R}_{\mathrm{CS}} = \left\{ ((\mathbf{w}_1, \ldots, \mathbf{w}_n, k), \mathbf{w}_0) \;\middle|\; \begin{array}{c} k \leq m, \\ \mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_n \in \mathbb{Z}_2^m, \\ \forall i \in [n], d(\mathbf{w}_0, \mathbf{w}_i) \leq k \end{array} \right\} .$$

This relation is NP-complete.

A Σ-protocol for $\mathcal{R}_{\mathrm{CS}}$ is described on the next page.

(i) Prove that the protocol is perfectly complete. [5 marks]

(ii) Prove that the protocol is SHVZK. [5 marks]

(iii) Prove that the protocol is computationally $5n$-special-sound, justifying the fact that the extractor's output is a witness for $\mathcal{R}_{\mathrm{CS}}$. [8 marks]

(iv) Look at the highlighted items in the protocol. Consider a modified version of the protocol where the verifier **does not** send $I \leftarrow_\$ \{1, \ldots, n\}$ to the prover, and when the verifier sends challenge $c = j$, the prover sends highlighted messages and decommitments **for all** $I \in [n]$, and the verifier checks line (j) **for all** $I \in [n]$.

Does your simulation strategy from part (ii) adapt to the modified protocol? Explain why or why not. [2 marks]

Notation

Let $\lambda \in \mathbb{N}$ denote the security parameter.

Let " $*$ " denote the restriction of a vector to its first $m - k$ entries.

Let $\Sigma_m \subset \mathbb{Z}_2^{m \times m}$ denote the set of permutation matrices for permuting m items.

Let $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ be a perfectly hiding and computationally binding commitment scheme for messages in $\mathbb{Z}_2$, with commitment and de-commitment spaces $\mathcal{C}$ and $\mathcal{D}$ respectively[1]. We will suppress pp for notational convenience and write e.g. $\mathsf{Commit}(\mathbf{a})$ and $\mathsf{Commit}(p)$ for vectors $\mathbf{a}$ and matrices p with the understanding that the commitment and verification algorithms are applied entry-wise.

[1] I.e., for any $\mathbf{u} \in \mathbb{Z}_2$, we have $(c, d) \leftarrow \mathsf{Commit}(pp, \mathbf{u})$ where $c \in \mathcal{C}$ and $d \in \mathcal{D}$.

Protocol Specification

Inputs

The prover receives $pp \leftarrow_\$ \mathsf{Setup}(1^\lambda)$ and $((\mathbf{w}_1, \ldots, \mathbf{w}_n, k), \mathbf{w}_0) \in \mathcal{R}_{\text{CS}}$.

The verifier receives pp and $(\mathbf{w}_1, \ldots, \mathbf{w}_n, k)$.

First Prover Message

Sample and compute the following values:

$$\begin{aligned} &\mathbf{a} \leftarrow_\$ \mathbb{Z}_2^m \ , && (A, \alpha) \leftarrow_\$ \mathsf{Commit}(\mathbf{a}) \ , \\ &\bar{\mathbf{a}} := \mathbf{w}_0 \oplus \mathbf{a} \in \mathbb{Z}_2^m \ , && (\bar{A}, \bar{\alpha}) \leftarrow_\$ \mathsf{Commit}(\bar{\mathbf{a}}) \ . \end{aligned}$$

For each $i \in [n]$, compute $p_i \in \Sigma_m$ such that the first $m-k$ entries of $p_i \cdot \mathbf{w}_i$ and $p_i \cdot \mathbf{w}_0$ are equal,[2] then sample and compute the following values:

$$\begin{aligned} &r_i \leftarrow_\$ \Sigma_m \ , && (R_i, \rho_i) \leftarrow_\$ \mathsf{Commit}(r_i) \ , \\ &\mathbf{b}_i := r_i \cdot \mathbf{a} \ , && (B_i, \beta_i) \leftarrow_\$ \mathsf{Commit}(\mathbf{b}_i) \ , \\ &\bar{\mathbf{b}}_i := r_i \cdot (\bar{\mathbf{a}} \oplus \mathbf{w}_i) \ , && (\bar{B}_i, \bar{\beta}_i) \leftarrow_\$ \mathsf{Commit}(\bar{\mathbf{b}}_i) \ , \end{aligned}$$

$$\begin{aligned} &s_i := p_i \cdot r_i^{-1} \in \Sigma_m \ , && (S_i, \sigma_i) \leftarrow_\$ \mathsf{Commit}(s_i) \ , \\ &\mathbf{d}_i := s_i \cdot \mathbf{b}_i \ , && (D_i, \delta_i) \leftarrow_\$ \mathsf{Commit}(\mathbf{d}_i) \ , \\ &\bar{\mathbf{d}}_i := s_i \cdot \bar{\mathbf{b}}_i \ , && (\bar{D}_i, \bar{\delta}_i) \leftarrow_\$ \mathsf{Commit}(\bar{\mathbf{d}}_i) \ . \end{aligned}$$

Send commitments $A, \bar{A}, \{R_i, B_i, \bar{B}_i, S_i, D_i, \bar{D}_i\}_{i=1}^n$ to the verifier.

Verifier Challenge

Sample $c \leftarrow_\$ \{1, \ldots, 5\}$ and $I \leftarrow_\$ \{1, \ldots, n\}$ and send them to the prover.

Prover Response and Verifier Checks

$c = 1$: The prover sends $\mathbf{a} \in \mathbb{Z}_2^m, \alpha \in \mathcal{D}^m$, $r_I \in \Sigma_m, \beta_I \in \mathcal{D}^m$ and $\rho_I \in \mathcal{D}^{m \times m}$.
The verifier accepts if $\mathsf{Verify}(A, \alpha, \mathbf{a}) = 1^m$ and

$$r_I \in \Sigma_m \ , \quad \mathsf{Verify}(R_I, \rho_I, r_I) = 1^{m \times m} \ , \text{ and } \quad \mathsf{Verify}(B_I, \beta_I, r_I \cdot \mathbf{a}) = 1^m \ . \tag{1}$$

$c = 2$: The prover sends $\bar{\mathbf{a}} \in \mathbb{Z}_2^m, \bar{\alpha} \in \mathcal{D}^m$, $r_I \in \Sigma_m$, $\bar{\beta}_I \in \mathcal{D}^m$ and $\rho_I \in \mathcal{D}^{m \times m}$.
The verifier accepts if $\mathsf{Verify}(\bar{A}, \bar{\alpha}, \bar{\mathbf{a}}) = 1^m$ and

$$r_I \in \Sigma_m, \quad \mathsf{Verify}(R_I, \rho_I, r_I) = 1^{m \times m} \ , \text{ and } \quad \mathsf{Verify}(\bar{B}_I, \bar{\beta}_I, r_I \cdot (\bar{\mathbf{a}} \oplus \mathbf{w}_I)) = 1^m \ . \tag{2}$$

$c = 3$: The prover sends $\mathbf{b}_I \in \mathbb{Z}_2^m$, $s_I \in \Sigma_m$, $\beta_I, \delta_I \in \mathcal{D}^m$ and $\sigma_I \in \mathcal{D}^{m \times m}$.
The verifier accepts if

$$s_I \in \Sigma_m \ , \mathsf{Verify}(B_I, \beta_I, \mathbf{b}_I) = 1^m \ , \mathsf{Verify}(S_I, \sigma_I, s_I) = 1^{m \times m} \ , \text{ and } \mathsf{Verify}(D_I, \delta_I, s_I \cdot \mathbf{b}_I) = 1^m \ . \tag{3}$$

$c = 4$: The prover sends $\bar{\mathbf{b}}_I \in \mathbb{Z}_2^m$, $s_I \in \Sigma_m$, $\bar{\beta}_I, \bar{\delta}_I \in \mathcal{D}^m$ and $\sigma_I \in \mathcal{D}^{m \times m}$.
The verifier accepts if

$$s_I \in \Sigma_m \ , \mathsf{Verify}(\bar{B}_I, \bar{\beta}_I, \bar{\mathbf{b}}_I) = 1^m \ , \mathsf{Verify}(S_I, \sigma_I, s_I) = 1^{m \times m} \ , \text{ and } \mathsf{Verify}(\bar{D}_I, \bar{\delta}_I, s_I \cdot \bar{\mathbf{b}}_I) = 1^m \ . \tag{4}$$

$c = 5$: The prover sends $\mathbf{d}_I^*, \bar{\mathbf{d}}_I^* \in \mathbb{Z}_2^{m-k}$ and $\delta_I^*, \bar{\delta}_I^* \in \mathcal{D}^{m-k}$. The verifier accepts if

$$\mathbf{d}_I^* \ \oplus \ \bar{\mathbf{d}}_I^* = 0^{m-k} \ , \quad \mathsf{Verify}(D_I^*, \delta_I^*, \mathbf{d}_I^*) = 1^{m-k} \ , \text{ and } \quad \mathsf{Verify}(\bar{D}_I^*, \bar{\delta}_I^*, \bar{\mathbf{d}}_I^*) = 1^{m-k} \ . \tag{5}$$

[2]Note p_i may not be unique.