CS 276: Homework 5

Due Date: Friday October 18th, 2024 at 8:59pm via Gradescope

1 Signature Scheme from CDH

We will construct a signature scheme that resembles the Schnorr signature scheme and prove it secure given the CDH assumption.

Let $\mathbb{G}$ be a cryptographic group of prime order p that is generated by g. Also, let p be super-polynomial in the security parameter λ. Let us also define two random oracles $H : \mathbb{G} \to \mathbb{G}$ and $G : \mathcal{M} \times \mathbb{G}^6 \to \mathbb{Z}_p$, where $\mathcal{M}$ is the message space.

1. $\mathsf{Gen}(1^\lambda)$: Sample $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $y = g^x$. Output $\mathsf{pk} = y$ and $\mathsf{sk} = x$.

2. $\mathsf{Sign}(\mathsf{sk}, m)$: To sign a message $m \in \mathcal{M}$, sample $k \xleftarrow{\$} \mathbb{Z}_p$ and compute the following:

$$\begin{aligned} u &= g^k \\ h &= H(u) \\ z &= h^{\mathsf{sk}} \\ v &= h^k \\ c &= G(m, g, h, \mathsf{pk}, z, u, v) \\ s &= k + c \cdot \mathsf{sk} \mod p \\ \sigma &= (z, s, c) \end{aligned}$$

   Output σ.

3. $\mathsf{Verify}(\mathsf{pk}, m, \sigma)$: Compute the following:

$$\begin{aligned} u' &= g^s \cdot \mathsf{pk}^{-c} \\ h' &= H(u') \\ v' &= h'^s \cdot z^{-c} \\ c' &= G(m, g, h', \mathsf{pk}, z, u', v') \end{aligned}$$

   Output 1 (accept) if $c = c'$ and 0 (reject) otherwise.

Definition 1.1 (Computational Diffie-Hellman (CDH) Assumption) *The CDH challenger samples $a, b \xleftarrow{\$} \mathbb{Z}_p$ independently and gives the adversary (g, g^a, g^b). The adversary wins the CDH game if they return $g^{a \cdot b}$. The CDH assumption states that for any PPT adversary, the probability that the adversary wins the CDH game is* $\mathsf{negl}(\lambda)$.

Question: Prove that the signature scheme constructed above is secure in the random oracle model given the CDH assumption.

Solution The solution is based on [CM05], section 4.

Given an adversary $\mathcal{A}_{Sign}$ that breaks the security of the signature scheme, we construct the following CDH adversary $\mathcal{A}_{CDH}$ that breaks breaks the CDH assumption.

Construction of $\mathcal{A}_{CDH}$:

1. $\mathcal{A}_{CDH}$ receives (g, g^a, g^b). Then $\mathcal{A}_{CDH}$ initializes the signing adversary $\mathcal{A}_{Sign}$ with security parameter 1^λ and $\mathsf{pk} = g^a$. That means implicitly, $\mathsf{sk} = a$.

2. **Simulated Random Oracle:** $\mathcal{A}_{CDH}$ keeps a truth table $\mathcal{H}$ for H and a truth table $\mathcal{G}$ for G, which works similarly.

   Initially, $\mathcal{H} = \{\}$, but $\mathcal{H}$ can be reprogrammed. If $(u, h) \in \mathcal{H}$, then $H(u) = h$. On the other hand, if for a given input u, there is no h such that $(u, h) \in \mathcal{H}$, then $H(u) = \perp$. Finally, each input $u \in \mathbb{G}$ can have at most one output, so there is at most one h-value such that $(u, h) \in \mathcal{H}$.

3. $\mathcal{A}_{CDH}$ runs $\mathcal{A}_{Sign}$ internally, and handles queries to $H, G, \mathsf{Sign}(\mathsf{sk}, \cdot)$ as follows.

   - $H(u)$: On input $u \in \mathbb{G}$:
     (a) If $H(u) = \perp$, then sample $d \xleftarrow{\$} \mathbb{Z}_p$, and append $(u, g^b \cdot g^d)$ to $\mathcal{H}$ so that now, $H(u) = g^b \cdot g^d$.
     (b) Return $H(u)$.
   - $G(m, g, h, \mathsf{pk}, z, u, v)$: On input $(m, g, h, \mathsf{pk}, z, u, v)$:
     (a) If $G(m, g, h, \mathsf{pk}, z, u, v) = \perp$, then sample $d \xleftarrow{\$} \mathbb{Z}_p$ and append $\big((m, g, h, \mathsf{pk}, z, u, v), d\big)$ to $\mathcal{G}$ so that $G(m, g, h, \mathsf{pk}, z, u, v) = d$.
     (b) Return $G(m, g, h, \mathsf{pk}, z, u, v)$.
   - $\mathsf{Sign}(\mathsf{sk}, m)$: On input $m \in \mathcal{M}$, do the following:
     (a) Sample $(\kappa, c, s) \xleftarrow{\$} \mathbb{Z}_p^3$.
     (b) Compute

$$\begin{aligned} u &= g^s \cdot \mathsf{pk}^{-c} \\ h &= g^\kappa \\ z &= \mathsf{pk}^\kappa \\ v &= h^s \cdot z^{-c} \\ \sigma &= (z, s, c) \end{aligned}$$

     (c) If $H(u) \neq \perp$, then $\mathcal{A}_{CDH}$ outputs $\perp$ and aborts. Otherwise, it appends (u, h) to $\mathcal{H}$. Likewise, if $G(m, g, h, \mathsf{pk}, z, u, v) \neq \perp$, then $\mathcal{A}_{CDH}$ outputs $\perp$ and aborts. Otherwise, it appends $\big((m, g, h, \mathsf{pk}, z, u, v), c\big)$ to $\mathcal{G}$.
     (d) Return σ.

4. When $\mathcal{A}_{Sign}$ outputs an attempted forgery $(m^*, (z^*, s^*, c^*))$, $\mathcal{A}_{CDH}$ checks that

$$\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$$

   and that $(m^*, (z^*, s^*, c^*))$ were not previously generated on a query to Sign. If at least one check fails, then $\mathcal{A}_{CDH}$ outputs $\perp$ and aborts. Otherwise, if both checks pass, then $\mathcal{A}_{CDH}$ computes:

$$u^* := g^{s^*} \cdot \mathsf{pk}^{-c^*}$$

   and continues.

5. We can assume that $H(u^*) \neq \bot$ because $\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$.

   (a) Case 1: If the value of $H(u^*)$ was determined during one of $\mathcal{A}_{Sign}$'s queries to H, then $\mathcal{A}_{CDH}$ looks up the value of d such that $H(u^*) = g^b \cdot g^d$. Then $\mathcal{A}_{CDH}$ computes and outputs:

   $$z^* \cdot (g^a)^{-d}$$

   as its guess for $g^{a \cdot b}$.

   (b) Case 2: If the value of $H(u^*)$ was determined during one of $\mathcal{A}_{Sign}$'s queries to $\mathsf{Sign}(\mathsf{sk}, \cdot)$, then $\mathcal{A}_{CDH}$ looks up the values of (m', c', s') from that query. Note that $u^* = g^{s'} \cdot \mathsf{pk}^{-c'}$. Then $\mathcal{A}_{CDH}$ computes and outputs:

   $$(g^b)^{(s^* - s')/(c^* - c')}$$

   as its guess for $g^{a \cdot b}$.

Analysis $\mathcal{A}_{CDH}$ correctly simulates the signature security game for $\mathcal{A}_{Sign}$. Assuming that $\mathcal{A}_{CDH}$ does not abort during the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$, $\mathcal{A}_{CDH}$ correctly simulates the oracles for $H, G, \mathsf{Sign}(\mathsf{sk}, \cdot)$ (lemma 1.3). Furthermore, the probability that $\mathcal{A}_{CDH}$ aborts during the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$ is negligible (lemma 1.2).

Next, $\mathcal{A}_{Sign}$ will output a valid forgery with non-negligible probability. This means that

$$\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$$

and m^* was not previously queried to Sign. Then $\mathcal{A}_{CDH}$ will reach either case 1 or 2.

Next, if $\mathcal{A}_{CDH}$ reaches cases 1 or 2, then $\mathcal{A}_{CDH}$ will compute the correct output with overwhelming probability. If $\mathcal{A}_{CDH}$ reaches case 1, then

$$g^{a \cdot b} = z^* \cdot (g^a)^{-d}$$

with overwhelming probability (lemma 1.4). If $\mathcal{A}_{CDH}$ reaches case 2, then

$$g^{a \cdot b} = (g^b)^{(s^* - s')/(c^* - c')}$$

with overwhelming probability (lemma 1.7).

Lemmas

Lemma 1.2 *The probability that $\mathcal{A}_{CDH}$ outputs $\bot$ and aborts during the simulation of* $\mathsf{Sign}(\mathsf{sk}, m)$ *is* $\mathsf{negl}(\lambda)$.

Proof. $\mathcal{A}_{CDH}$ outputs $\bot$ and aborts during the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$ if $H(u)$ or $G(m, g, h, \mathsf{pk}, z, u, v)$ already have a value determined from previous steps.

u is uniformly random and independent of all variables in previous rounds. This is because

$$u = g^s \cdot \mathsf{pk}^{-c}$$

where s is uniformly random in $\mathbb{Z}_p$ and independent of all previously computed variables.

At any point in the simulation, $\mathcal{H}$ contains $\mathsf{poly}(\lambda)$-many input-output pairs. The probability that $(u, *) \in \mathcal{H}$ is $\mathsf{poly}(\lambda)/|\mathbb{G}| = \mathsf{negl}(\lambda)$. Then in the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$, the probability that $H(u) \neq \bot$ is $\mathsf{negl}(\lambda)$.

Likewise for $G(m, g, h, \mathsf{pk}, z, u, v)$: there are $|\mathbb{G}|$ possible values that u can take and all are equally likely, over the randomness of s. $\mathcal{G}$ contains $\mathsf{poly}(\lambda)$-many input-output pairs. The probability that $\big((m, g, h, \mathsf{pk}, z, u, v), *\big) \in \mathcal{G}$ is $\mathsf{poly}(\lambda)/|\mathbb{G}| = \mathsf{negl}(\lambda)$. Then in the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$, the probability that $G(m, g, h, \mathsf{pk}, z, u, v) \neq \bot$ is $\mathsf{negl}(\lambda)$.

Lemma 1.3 *Given that $\mathcal{A}_{CDH}$ does not abort during the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$, $\mathcal{A}_{CDH}$ correctly simulates the oracles for $H, G, \mathsf{Sign}(\mathsf{sk}, \cdot)$.*

Proof. First, $(g, \mathsf{pk}, \mathsf{sk})$ have the correct distribution. $\mathsf{sk} = a$, which is uniformly random in $\mathbb{Z}_p$, and $\mathsf{pk} = g^{\mathsf{sk}}$.

Second, H is simulated correctly because each query to H receives a uniformly random response that is independent of the output of H on any other input. When $\mathcal{A}_{Sign}$ queries H, they receive the response $g^b \cdot g^d$, which is uniformly random due to the randomness of d. In the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$, the value of $H(u)$ is reprogrammed to $h = g^{\kappa}$, which is uniformly random due to the randomness of κ.

Third, G is simulated correctly because each query to G receives a uniformly random response that is independent of the output of G on any other input. When $\mathcal{A}_{Sign}$ queries G, they receive the response d, which is uniformly random. In the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$, the value of $G(m, g, h, \mathsf{pk}, z, u, v)$ is reprogrammed to c which is uniformly random.

Fourth, the variables

$$(u, h, z, v, c, s)$$

have the same distribution in the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$ as they do in the real signature game. In the real signature game:

- c is uniformly random because it is the output of $G(m, g, h, \mathsf{pk}, z, u, v)$, and with overwhelming probability, G has not previously been queried on $(m, g, h, \mathsf{pk}, z, u, v)$.
- s is uniformly random due to the randomness of k. Recall that $s = k + c \cdot \mathsf{sk} \mod p$.
- h is uniformly random because it is the output of $H(u)$, and with overwhelming probability, H has not previously been queried on u.
- Given $(c, s, h, \mathsf{pk}, \mathsf{sk})$, the variables (u, z, v) are completely determined by the following equations:

$$u = g^s \cdot \mathsf{pk}^{-c} \tag{1.1}$$

$$z = h^{\mathsf{sk}} = g^{\log_g(h) \cdot \mathsf{sk}} = \left(g^{\mathsf{sk}}\right)^{\log_g(h)} \tag{1.2}$$

$$= \mathsf{pk}^{\log_g(h)} \tag{1.3}$$

$$v = h^s \cdot z^{-c} \tag{1.4}$$

In the simulation of $\mathsf{Sign}(\mathsf{sk}, m)$:

- c and s are uniformly random and independent. Also, h is uniformly random due to the randomness of κ.

- Given $(c, s, h, \mathsf{pk}, \mathsf{sk})$, the variables (u, z, v) are completely determined by the same equations – 1.1, 1.3, 1.4 – as in the real signature game.

Lemma 1.4 *If $\mathcal{A}_{CDH}$ reaches case 1, then with overwhelming probability:*

$$g^{a \cdot b} = z^* \cdot (g^a)^{-d}$$

Proof. Recall that $\mathcal{A}$'s output is $(m^*, (z^*, s^*, c^*))$, and let the variables computed by $\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*))$ be the following:

$$\begin{aligned} u' &= g^{s^*} \cdot \mathsf{pk}^{-c^*} \\ h' &= H(u') \\ v' &= h'^{s^*} \cdot (z^*)^{-c^*} \\ c' &= G(m^*, g, h', \mathsf{pk}, z^*, u', v') \end{aligned}$$

Next, lemma 1.5 shows that the probability that $\mathcal{A}$ outputs an $(m^*, (z^*, s^*, c^*))$ such that $\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$ but $\log_g(\mathsf{pk}) \neq \log_{h'}(z^*)$ is negligible. So from now on, let us assume that $\log_g(\mathsf{pk}) = \log_{h'}(z^*)$. Then:

$$\begin{aligned} z^* &= h'^{\log_g(\mathsf{pk})} = g^{(b+d) \cdot a} = g^{a \cdot b + a \cdot d} \\ z^* \cdot (g^a)^{-d} &= g^{a \cdot b} \end{aligned}$$

Lemma 1.5 *The probability that $\mathcal{A}$ outputs an $(m^*, (z^*, s^*, c^*))$ such that $\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$ but $\log_g(\mathsf{pk}) \neq \log_{h'}(z^*)$ is negligible.*

Proof. $\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$ only if c' satisfies $u' = g^{s^*} \cdot \mathsf{pk}^{-c'}$ and $v' = h'^{s^*} \cdot (z^*)^{-c'}$. However, the value of $c' = G(m^*, g, h', \mathsf{pk}, z^*, u', v')$ is sampled uniformly at random *after* $(m^*, g, h', \mathsf{pk}, z^*, u', v')$ have been fixed.

For any $(m^*, g, h', \mathsf{pk}, z^*, u', v')$, if $\log_g(\mathsf{pk}) \neq \log_{h'}(z^*)$, then there is at most one value of (s^*, c') such that $u' = g^{s^*} \cdot \mathsf{pk}^{-c'}$ and $v' = h'^{s^*} \cdot (z^*)^{-c'}$ (lemma 1.6).

With overwhelming probability, each query $(m^*, g, h', \mathsf{pk}, z^*, u', v')$ to G for which $\log_g(\mathsf{pk}) \neq \log_{h'}(z^*)$ will result in a c' such that $u' \neq g^{s^*} \cdot \mathsf{pk}^{-c'}$ or $v' \neq h'^{s^*} \cdot (z^*)^{-c'}$. In this case, there is no value of c^* for which $\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$.

Since $\mathcal{A}$ is limited to making only polynomially-many queries to G, $\mathcal{A}$ has negligible probability of finding a $(m^*, (z^*, s^*, c^*))$ such that $\mathsf{Verify}(\mathsf{pk}, m^*, (z^*, s^*, c^*)) = 1$ but $\log_g(\mathsf{pk}) \neq \log_{h'}(z^*)$.

Lemma 1.6 *For a given $(m, g, h, \mathsf{pk}, z, u, v)$, if $\log_g(\mathsf{pk}) \neq \log_h(z)$, then there is at most one value of (s, c) for which $u = g^s \cdot \mathsf{pk}^{-c}$ and $v = h^s \cdot z^{-c}$.*

Proof. Let $\mathsf{sk} = \log_g(\mathsf{pk})$ and let $\mathsf{sk}' = \log_h(z)$. Also, let $k = \log_g(u)$ and let $k' = \log_h(v)$. Then

$$\begin{aligned} g^s \cdot \mathsf{pk}^{-c} &= g^{s - c \cdot \mathsf{sk}} \\ h^s \cdot z^{-c} &= h^{s - c \cdot \mathsf{sk}'} \end{aligned}$$

Next,

$$u = g^s \cdot \mathsf{pk}^{-c} \iff k = s - c \cdot \mathsf{sk}$$
$$v = h^s \cdot z^{-c} \iff k' = s - c \cdot \mathsf{sk}'$$

If $\mathsf{sk} \neq \mathsf{sk}'$, then the only way that $u = g^s \cdot \mathsf{pk}^{-c}$ and $v = h^s \cdot z^{-c}$ is if

$$c = \frac{k - k'}{\mathsf{sk}' - \mathsf{sk}} \text{ and } s = k + c \cdot \mathsf{sk} \tag{1.5}$$

Lemma 1.7 *If $\mathcal{A}_{CDH}$ reaches case 2, then with overwhelming probability:*

$$g^{a \cdot b} = (g^b)^{(s^* - s')/(c^* - c')}$$

Proof. In case 2,

$$u^* = g^{s^*} \cdot \mathsf{pk}^{-c^*} = g^{s'} \cdot \mathsf{pk}^{-c'}$$

If $c^* \neq c'$, then

$$\mathsf{pk} = g^a = g^{(s^* - s')/(c^* - c')}$$
$$a = \frac{s^* - s'}{c^* - c'}$$
$$g^{a \cdot b} = \left(g^b\right)^{(s^* - s')/(c^* - c')}$$

It just remains to show that $c^* \neq c'$. Since only polynomially-many queries are made to G, with overwhelming probabiliy over the randomness of G, every distinct query to G produces a unique output value. We also know that m^* was not previously queried to $\mathsf{Sign}(\mathsf{sk}, \cdot)$, so $m^* \neq m'$. Since

$$c^* = G(m^*, g, h^*, \mathsf{pk}, z^*, u^*, v^*)$$
$$c' = G(m', g, h', \mathsf{pk}, z', u', v')$$

then $c^* \neq c'$ with overwhelming probability. ■

2 Additively Homomorphic Encryption (AHE)

Some natural encryption schemes, such as El Gamal encryption, are additively homomorphic[1], meaning that $\mathsf{Enc}(m^{(1)})$ and $\mathsf{Enc}(m^{(2)})$ can be combined into a valid encryption of $m^{(1)} + m^{(2)}$ without knowledge of the secret key. It turns out that this property is sufficient to construct public-key encryption. We will show that secret-key additively homomorphic encryption implies public-key encryption.

[1]This is assuming we use the additive notation for operations over the cryptographic group.

Definition 2.1 (Additively Homomorphic Encryption) *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, H_\oplus)$ *be four PPT algorithms with message space* $\mathcal{M} = \{0,1\}$ *and ciphertext space* $\mathcal{C}$*. Let* $H_\oplus$ *map* $\mathcal{C}^\ell \to \mathcal{C}$*, for any* $\ell = \mathsf{poly}(\lambda)$*.*

Next, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, H_\oplus)$ *is a* ***secret-key additively homomorphic encryption (AHE) scheme***[2] *if the following properties are satisfied:*

- ***Perfect Correctness:*** *For any* $\ell = \mathsf{poly}(\lambda)$ *messages* $(m^{(1)}, \ldots, m^{(\ell)}) \in \{0,1\}^\ell$*:*

$$\Pr\left[\mathsf{Dec}\Big(\mathsf{sk}, H_\oplus\big[\mathsf{Enc}(\mathsf{sk}, m^{(1)}), \ldots, \mathsf{Enc}(\mathsf{sk}, m^{(\ell)})\big]\Big) = \sum_{i \in [\ell]} m^{(i)} \mod 2\right] = 1$$

- ***Compactness:*** *There exists a polynomial function* $m(\cdot)$ *such that for any* $\ell = \mathsf{poly}(\lambda)$ *messages* $(m^{(1)}, \ldots, m^{(\ell)}) \in \{0,1\}^\ell$*, the length of* $H_\oplus\big[\mathsf{Enc}(\mathsf{sk}, m^{(1)}), \ldots, \mathsf{Enc}(\mathsf{sk}, m^{(\ell)})\big]$ *is upper-bounded by* $m(\lambda)$*.*[3]

- ***CPA security:*** $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *constitute a CPA secure encryption scheme.*

The following construction builds a public-key encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ from a secret-key AHE scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, H_\oplus)$.

1. $\mathsf{Gen}'(1^\lambda)$: Compute the following:

$$\begin{aligned} &\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda) \\ &\ell' = 4m(\lambda) \\ &r \xleftarrow{\$} \{0,1\}^{\ell'} \backslash \{0^{\ell'}\} \\ &X_i \leftarrow \mathsf{Enc}(\mathsf{sk}, r_i), \quad \forall i \in [\ell'] \\ &\mathsf{pk} = (X_1, \ldots, X_{\ell'}, r) \end{aligned}$$

   Then output $(\mathsf{pk}, \mathsf{sk})$.

2. $\mathsf{Enc}'(\mathsf{pk}, m)$:
   (a) Sample $s \in \{0,1\}^{\ell'}$ uniformly at random such that $\langle r, s\rangle = m$.[4]
   (b) Let X_s be a tuple of all the X_i-values for which $s_i = 1$.
   (c) Compute and output $c = H_\oplus(X_s)$.

3. $\mathsf{Dec}'(\mathsf{sk}, c)$: Output $\mathsf{Dec}(\mathsf{sk}, c)$.

[2] *Public-key* additively homomorphic encryption is defined similarly, except $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ are a public-key encryption scheme, $H_\oplus$ takes pk as input, and Enc takes pk, instead of sk, as input.

[3] Note that $m(\lambda)$ is independent of ℓ.

[4] $\langle r, s\rangle = \sum_{i \in [\ell']} r_i \cdot s_i \mod 2$. We can sample s using rejection sampling: sample $s \xleftarrow{\$} \{0,1\}^{\ell'}$ and check whether $\langle r, s\rangle = m$. If not, then reject this s and repeat the procedure.

Question: Prove that if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, H_\oplus)$ is a secret-key AHE scheme, then $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ satisfies (1) CPA security and (2) the following notion of perfect correctness:

$$\Pr\left[\mathsf{Dec}'(\mathsf{sk}, \mathsf{Enc}'(\mathsf{pk}, m)) = m\right] = 1, \quad \forall m \in \{0,1\}$$

Solution This proof is based on [Rot11].

Lemma 2.2 $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *satisfies perfect correctness.*

Proof. For any message $m \in \{0,1\}$, let $c = \mathsf{Enc}'(\mathsf{pk}, m)$. Then there exists some $s \in \{0,1\}^{\ell'}$ such that $\langle r, s\rangle = m$ and $c = H_\oplus(X_s)$.

Then:

$$\begin{aligned}
\mathsf{Dec}'\left[\mathsf{sk}, \mathsf{Enc}'(\mathsf{pk}, m)\right] &= \mathsf{Dec}\big(\mathsf{sk}, H_\oplus(X_s)\big) \\
&= \mathsf{Dec}\Big(\mathsf{sk}, H_\oplus\big[\big(\mathsf{Enc}(\mathsf{sk}, r_i)\big)_{\forall i \in [\ell'] : s_i = 1}\big]\Big) \\
&= \sum_{i \in [\ell'] : s_i = 1} r_i \mod 2 \\
&= \sum_{i \in [\ell']} r_i \cdot s_i \mod 2 = \langle r, s\rangle \\
&= m
\end{aligned}$$

Therefore, $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ satisfies perfect correctness.

Lemma 2.3 $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *satisfies CPA security.*

Proof. Consider the following sequence of hybrids:

- $\mathcal{H}_0$: The CPA security game for $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$. Without loss of generality, we can assume that the adversary's challenge messages are $m_0 = 0$ and $m_1 = 1$.
  1. **Setup:** The challenger computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}'(1^\lambda)$ and sends pk to $\mathcal{A}$.
  2. **Challenge:** The adversary submits messages $m_0 = 0$ and $m_1 = 1$. The challenger samples $b \leftarrow \{0,1\}$ and computes $c = \mathsf{Enc}'(\mathsf{pk}, m_b)$ as follows:

     They sample $s \xleftarrow{\$} \{s' \in \{0,1\}^{\ell'} : m_b = \langle r, s'\rangle\}$ and compute $c = H_\oplus(X_s)$.[5]
     Then they send c to $\mathcal{A}$.
  3. **Response:** $\mathcal{A}$ responds with $b' \in \{0,1\}$. The output of the hybrid is 1 if $b = b'$ and 0 otherwise.

- $\mathcal{H}_1$: Same as $\mathcal{H}_0$, except for all $i \in [\ell']$, $X_i = \mathsf{Enc}(\mathsf{sk}, r_i)$ is replaced with

$$X_i' = \mathsf{Enc}(\mathsf{sk}, 0)$$

[5]Note that for each $b \in \{0,1\}$, $m_b = b$.

- $\mathcal{H}_2$: Same as $\mathcal{H}_1$, except instead of sampling $b \xleftarrow{\$} \{0,1\}$ and then sampling $s \xleftarrow{\$} \{s' \in \{0,1\}^{\ell'} : m_b = \langle r, s' \rangle\}$, the challenger first samples $s \xleftarrow{\$} \{0,1\}^{\ell'}$ and then computes $b = m_b = \langle r, s \rangle$.

- $\mathcal{H}_3$: Same as $\mathcal{H}_1$, except instead of sampling $r \xleftarrow{\$} \{0,1\}^{\ell'} \backslash \{0^{\ell'}\}$ and $s \xleftarrow{\$} \{0,1\}^{\ell'}$, the challenger samples $r \xleftarrow{\$} \{0,1\}^{\ell'}$ and $s \xleftarrow{\$} \{0,1\}^{\ell'} \backslash \{0^{\ell'}\}$.

Claim 2.4 $\big|\Pr[\mathcal{H}_0 \to 1] - \Pr[\mathcal{H}_3 \to 1]\big| = \mathsf{negl}(\lambda)$

Proof. $\mathcal{H}_0$ and $\mathcal{H}_1$ are indistinguishable due to the CPA security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

$\mathcal{H}_1$ and $\mathcal{H}_2$ are perfectly indistinguishable because the only difference between the two hybrids is the order in which b and s are sampled, but the joint distribution of (b, s) is the same in both hybrids.

Since $r \neq 0^{\ell'}$, then $\langle r, s \rangle = 0$ for exactly half of the s-values in $\{0,1\}^{\ell'}$, and $\langle r, s \rangle = 1$ for the other half. Therefore, if s is sampled uniformly at random from $\{0,1\}^{\ell'}$, then $b = \langle r, s \rangle$ will be uniformly random over $\{0,1\}$ due to the randomness of s.

$\mathcal{H}_2$ and $\mathcal{H}_3$ are statistically indistinguishable because the distribution of (r, s) in the two hybrids is statistically close.

Then

$$\big|\Pr[\mathcal{H}_0 \to 1] - \Pr[\mathcal{H}_3 \to 1]\big| = \mathsf{negl}(\lambda)$$

Claim 2.5 $\Pr[\mathcal{H}_3 \to 1] = \frac{1}{2} + \mathsf{negl}(\lambda)$

Proof. We will use the leftover hash lemma to show that from the adversary's view in $\mathcal{H}_3$, b is statistically close to uniformly random.

First, let us define a hash function h_r:

$$h_r(s) = \langle r, s \rangle$$

where $r \xleftarrow{\$} \{0,1\}^{\ell'}$ and $s \in \{0,1\}^{\ell'} \backslash \{0^{\ell'}\}$. We claim that h_r is pairwise-independent.

Second, in $\mathcal{H}_3$, the variables (X', r, s, c, b) are sampled as follows:

$$\begin{aligned} X' &= (X'_1, \ldots, X'_{\ell'}) = (\mathsf{Enc}(\mathsf{sk}, 0), \ldots, \mathsf{Enc}(\mathsf{sk}, 0)) \\ r &\xleftarrow{\$} \{0,1\}^{\ell'} \\ s &\xleftarrow{\$} \{0,1\}^{\ell'} \backslash \{0^{\ell'}\} \\ c &= H_{\oplus}(X'_s) \\ b &= h_r(s) \end{aligned}$$

The adversary receives (X', c, r) and is asked to guess $h_r(s)$. Given (X', c), the variables (r, s) are uniformly random over $\{0,1\}^{\ell'} \times S_{X',c}$, where:

$$S_{X',c} = \{s' \in \{0,1\}^{\ell'} \backslash \{0^{\ell'}\} : c = H_{\oplus}(X'_s)\}$$

By the leftover hash lemma (lemma 2.6), for $b^* \xleftarrow{\$} \{0,1\}$, the statistical distance between

$$\big(X', c, r, h_r(s)\big) \quad \text{and} \quad (X', c, r, b^*)$$

is $2\sqrt{\frac{2}{|S_{X',c}|}}$.

Third,

$$\begin{aligned}
\Pr[\mathcal{H}_3 \to 1] = \Pr_{X',c,r,s}[\mathcal{A}(X', c, r) \to h_r(s)] &= \mathbb{E}_{X',c}\left[\Pr_{r,s}[\mathcal{A}(X', r, c) \to h_r(s) | X', c]\right] \\
&= \mathbb{E}_{X'}\left[\sum_c \Pr_s(c = H_\oplus(X'_s)|X') \cdot \Pr_{r,s}[\mathcal{A}(X', r, c) \to h_r(s)|X', c]\right] \\
&= \mathbb{E}_{X'}\left[\sum_c \frac{|S_{X',c}|}{2^{\ell'}-1} \cdot \Pr_{r,s}[\mathcal{A}(X', r, c) \to h_r(s)|X', c]\right] \\
&\le \mathbb{E}_{X'}\left[\sum_c \frac{|S_{X',c}|}{2^{\ell'-1}} \cdot \left(\Pr_{r,s,b^*}[\mathcal{A}(X', r, c) \to b^*|X', c] + 2\sqrt{\frac{2}{|S_{X',c}|}}\right)\right] \\
&= \mathbb{E}_{X'}\left[\sum_c \frac{|S_{X',c}|}{2^{\ell'-1}} \cdot \left(\frac{1}{2} + 2\sqrt{\frac{2}{|S_{X',c}|}}\right)\right] \\
&= \frac{1}{2} + \mathbb{E}_{X'}\left[\sum_c 2^{-(\ell'-1)} \cdot 2\sqrt{2} \cdot \sqrt{|S_{X',c}|}\right] \\
&\le \frac{1}{2} + 2\sqrt{2} \cdot 2^{-(\ell'-1)} \cdot \mathbb{E}_{X'}\left[\sum_c 2^{\ell'/2}\right] \\
&\le \frac{1}{2} + 2\sqrt{2} \cdot 2^{-(\ell'-1)} \cdot \mathbb{E}_{X'}\left[2^m \cdot 2^{\ell'/2}\right] = \frac{1}{2} + 2\sqrt{2} \cdot 2^{m-\ell'/2+1} \\
&= \frac{1}{2} + 4\sqrt{2} \cdot 2^{m-2m} = \frac{1}{2} + 4\sqrt{2} \cdot 2^{-m} \\
&= \frac{1}{2} + \mathsf{negl}(\lambda)
\end{aligned}$$

Lemma 2.6 (Leftover Hash Lemma) *Let h_r be a pairwise-independent hash function with a single-bit output. For a given subset S of the domain of h_r, let $r \xleftarrow{\$} \{0,1\}^{\ell'}$, $s \xleftarrow{\$} S$, and $b^* \xleftarrow{\$} \{0,1\}$. Then the statistical distance between*

$$\big(r, h_r(s)\big) \quad \textit{and} \quad (r, b^*)$$

is $2\sqrt{\frac{2}{|S|}}$.

A version of this lemma is stated in [Rot11], footnote 7, and [Gol08], theorem D.5.

Putting together the previous claims, we have that $\Pr[\mathcal{H}_0 \to 1] \le \frac{1}{2} + \mathsf{negl}(\lambda)$. Since $\mathcal{H}_0$ is the CPA security game, this shows that $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ satisfies CPA security.

■

References

[CM05] Benoît Chevallier-Mames. An efficient cdh-based signature scheme with a tight security reduction. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, pages 511–526, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, USA, 1 edition, 2008.

[Rot11] Ron Rothblum. Homomorphic encryption: From private-key to public-key. In Yuval Ishai, editor, *Theory of Cryptography*, pages 219–234, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.