# CS/ECE407 Cryptography – Homework 4
# Hash Functions and Public Key Cryptography

Due: Thursday, April 16, 3:30PM CT

Remember, you are free to collaborate with up to one classmate. See the course webpage for more details. You are expected to write out and submit your own solutions! Your collaboration is for discussing problems at a high level, not plagiarizing answers.

Your solutions should be typed or carefully handwritten. We provide a LaTeX template for this document, if you would like to use it as a starting point. **If we cannot read your handwritten answers, they will not receive credit.**

Typed solutions should be submitted through Gradescope (see course webpage). Hand-written solutions should be scanned and turned in through Gradescope.

**Definition 1** (Collision Resistance)**.** A hash function $H$ with seeds from $S$ is **collision resistant** if for any polytime adversary $\mathcal{A}$, the following probability is negligible:

$$\Pr\left[ H(s, x_0) = H(s, x_1) \;\middle|\; \begin{array}{l} s \leftarrow S(1^\lambda) \\ (x_0, x_1) \leftarrow \mathcal{A}(1^\lambda, s) \end{array} \right] < \text{negl}(\lambda)$$

**Definition 2** (Discrete Logarithm Hardness)**.** Let $\mathbb{G}_\lambda$ denote a family of cyclic groups with respective generators $g_\lambda$. We say that the discrete logarithm problem for this family is hard if for any polytime adversary $\mathcal{A}$, the following probability is negligible:

$$\Pr\left[ x = y \;\middle|\; \begin{array}{l} q = |\mathbb{G}_\lambda| \\ x \leftarrow \mathbb{Z}_q \\ y \leftarrow \mathcal{A}(1^\lambda, g_\lambda^x) \end{array} \right] < \text{negl}(\lambda)$$

Informally, we sample a random exponent $x$, give $g^x$ to the adversary, and the adversary wins if it outputs $x$.

**Problem 1** (Hash Functions)**.** Let $h : \{0,1\}^\lambda \times \{0,1\}^{2\lambda} \to \{0,1\}^\lambda$ be a collision resistant hash function whose seeds are uniformly sampled from $\{0,1\}^\lambda$. Consider the following function $H : \{0,1\}^\lambda \times \{0,1\}^{3\lambda} \to \{0,1\}^\lambda$ which splits input $x$ into three length-$\lambda$ strings, then calls $h$ twice:

$$\begin{array}{l} H(s, x): \\ \quad (x_0, x_1, x_2) = \mathsf{thirds}(x) \\ \quad \mathsf{return}\ h(s, h(s, x_0 \mid x_1) \mid x_2) \end{array}$$

1. **(2 points)** $H$ is also collision resistant. Give a security reduction that proves this.
2. **(2 points)** $H$ is well-defined only for inputs $x$ of length exactly $3\lambda$. Use $h$ to construct a hash function $H'$ that is well-defined for any input $x$ that has length *strictly less* than $3\lambda$. Give a security reduction that proves your $H'$ is collision resistant.
3. **(2 points)** Now, use $h$ to construct a collision-resistant function $H''$ that is well-defined for inputs $x$ of *any* polynomial length. Give one or two sentences that informally argument why your $H''$ is collision resistant; you do not need to prove this formally.

**Problem 2** (Discrete Logarithms and Collision Resistance)**.** Let $\mathbb{G}_\lambda, g_\lambda$ denote a family of discrete-log-hard groups (see Definition 2). Now, consider the following hash function definition. First, its seeds are drawn as follows:

$$S(1^\lambda) = \left\{ (g, h) \;\middle|\; \begin{array}{l} q \leftarrow |\mathbb{G}_\lambda| \\ g \leftarrow g_\lambda \\ x \leftarrow \mathbb{Z}_q \\ h \leftarrow g^x \end{array} \right\}$$

That is, we choose a group depending on $\lambda$, then sample a random group element $h$. Now, the hash function $H$ takes a seed $(g, h)$ and two integers $(a, b)$ and outputs a single group element as follows:

$$H((g, h), (a, b)) = g^a h^b$$

- **(3 points)** Give a security reduction showing that if the discrete logarithm problem is hard for groups $\mathbb{G}_\lambda$, then $H$ is a collision resistant hash function. Hint: *Remember, this means that if $H$ is* not *collision resistant—i.e., if there exists an efficient adversary that finds collisions with noticeable probability—then there should also be an adversary that computes discretes logarithms.*

**Problem 3** (Public Key Encryption)**.**

1. **(2 points)** It is crucial to use randomized encryption when using public keys. Suppose that we use a deterministic public-key encryption algorithm $Enc(pk, m)$. Describe an adversary that can decrypt a ciphertext from any such scheme in time that scales linearly in $|\mathcal{M}|$ where $\mathcal{M}$ is the scheme's message space.

2. **(3 points)** Consider the following attempt at key exchange:

   (a) Alice samples random key $k \leftarrow \{0,1\}^\lambda$ and random mask $r \leftarrow \{0,1\}^\lambda$. Alice sends $s = k \oplus r$ to Bob.

   (b) Bob samples $t \leftarrow \{0,1\}^\lambda$ and sends $u = t \oplus s$ to Alice.

   (c) Alice computes $w = u \oplus r$ and sends $w$ to Bob.

   (d) Alice outputs $k$; Bob outputs $w \oplus t$.

   Is the protocol correct? Namely do Alice and Bob compute the same key? Is the protocol secure? Why/why not?