6.5610 Recitation 1: Review

Katherine Zhao

February 9, 2024

Outline

- Hashing (OWF, Collision Resistance)
- AES
- Linear Algebra Review

Hashing

Hash Functions

A **Hash Function** $H : \{0,1\}^* \to \{0,1\}^\lambda$ maps strings from arbitrary length to strings of length λ

Useful properties for hash functions: collision resistance and one-wayness.

Applications for Hash Functions

Password Storage: Suppose a server wants to store the passwords of its users. However, we don't want to store the passwords directly.

Applications for Hash Functions

Password Storage: Suppose a server wants to store the passwords of its users. However, we don't want to store the passwords directly.

- Store $H(pw)$ instead

Applications for Hash Functions

Password Storage: Suppose a server wants to store the passwords of its users. However, we don't want to store the passwords directly.

- Store $H(pw)$ instead
- When a user logs in, checks that the hash of the input matches the stored hash value.

Applications for Hash Functions

Password Storage: Suppose a server wants to store the passwords of its users. However, we don't want to store the passwords directly.

- Store $H(pw)$ instead
- When a user logs in, checks that the hash of the input matches the stored hash value.
- Even if an adversary gets the stored hash values, we don't want them to discover the passwords of the users: **Use one-way functions!**

One-way Functions

Intuition

A polynomial time function H is said to be one-way if given $H(x)$ it is difficult to find x' such that $H(x) = H(x')$.

One-way Functions

Intuition

A polynomial time function H is said to be one-way if given $H(x)$ it is difficult to find x' such that $H(x) = H(x')$.

Formal Definition

A polynomial time function $H : \{0,1\}^* \to \{0,1\}^*$ is a *one-way function* (OWF) if for any probabilistic polynomial-time adversary A there exists a negligible function μ such that for every security parameter $\lambda \in \mathbb{N}$,

$$\Pr\left[H(x) = H(x') : \begin{array}{c} x \xleftarrow{R} \{0,1\}^\lambda \\ x' \leftarrow A(H(x)) \end{array} \right] \leq \mu(\lambda).$$

One-way Functions

Formal Definition

A polynomial time function $H : \{0,1\}^* \to \{0,1\}^*$ is a *one-way function* (OWF) if for any probabilistic polynomial-time adversary A there exists a negligible function μ such that for every security parameter $\lambda \in \mathbb{N}$,

$$\Pr\left[H(x) = H(x') : \begin{array}{c} x \xleftarrow{\mathrm{R}} \{0,1\}^\lambda \\ x' \leftarrow A(H(x)) \end{array}\right] \leq \mu(\lambda).$$

- λ is called the security parameter. The adversary and the function runs polynomial time in λ

One-way Functions

Formal Definition

A polynomial time function $H : \{0,1\}^* \to \{0,1\}^*$ is a *one-way function* (OWF) if for any probabilistic polynomial-time adversary A there exists a negligible function μ such that for every security parameter $\lambda \in \mathbb{N}$,

$$\Pr\left[H(x) = H(x') : \begin{array}{c} x \stackrel{R}{\leftarrow} \{0,1\}^\lambda \\ x' \leftarrow A(H(x)) \end{array} \right] \leq \mu(\lambda).$$

- λ is called the security parameter. The adversary and the function runs polynomial time in λ
- μ is a negligible function: for every polynomial p, there exists λ_0 such that for every $\lambda > \lambda_0$, $\mu(\lambda) < \frac{1}{p(\lambda)}$

One-way Functions

Formal Definition

A polynomial time function $H : \{0,1\}^* \to \{0,1\}^*$ is a *one-way function* (OWF) if for any probabilistic polynomial-time adversary A there exists a negligible function μ such that for every security parameter $\lambda \in \mathbb{N}$,

$$\Pr\left[H(x) = H(x') : \begin{array}{c} x \xleftarrow{\mathrm{R}} \{0,1\}^\lambda \\ x' \leftarrow A(H(x)) \end{array}\right] \le \mu(\lambda).$$

- λ is called the security parameter. The adversary and the function runs polynomial time in λ
- μ is a negligible function: for every polynomial p, there exists λ_0 such that for every $\lambda > \lambda_0$, $\mu(\lambda) < \frac{1}{p(\lambda)}$
- In practice, negligible is considered less than a very small constant, like 2^{-128}

Applications for Hash Functions

Authenticating Files: suppose a user wants to store a large file F on an untrusted server. We want to make sure that the server does not tamper with the file.

Applications for Hash Functions

Authenticating Files: suppose a user wants to store a large file F on an untrusted server. We want to make sure that the server does not tamper with the file.

- User stores a succinct hash $H(F)$ locally

Applications for Hash Functions

Authenticating Files: suppose a user wants to store a large file F on an untrusted server. We want to make sure that the server does not tamper with the file.

- User stores a succinct hash $H(F)$ locally
- When the user wants to use the file, it will fetch it from the server and receive F'

Applications for Hash Functions

Authenticating Files: suppose a user wants to store a large file F on an untrusted server. We want to make sure that the server does not tamper with the file.

- User stores a succinct hash $H(F)$ locally
- When the user wants to use the file, it will fetch it from the server and receive F'
- User can ensure integrity by checking if $H(F) = H(F')$: **Need collision resistance!**

Collision Resistance

Intuition

A hash function H is said to be collision resistant if it is hard to find x, x' such that $x \neq x'$ and $H(x) = H(x')$

Collision Resistance

Intuition

A hash function H is said to be collision resistant if it is hard to find x, x' such that $x \neq x'$ and $H(x) = H(x')$

Formal Definition

A family of functions $\{H_\lambda\}_{\lambda \in \mathbb{N}}$ where $H_\lambda : \{0,1\}^* \to \{0,1\}^\lambda$ is said to be *collision resistant* if for all polynomial-time adversaries A there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[H_\lambda(x) = H_\lambda(x') \wedge x \neq x' \;:\; (x, x') \leftarrow A(1^\lambda)\right] \leq \mu(\lambda).$$

Sample Problems

Let $f : \{0,1\}^n \to \{0,1\}^\lambda$ be a OWF. Define $g : \{0,1\}^{n+1} \to \{0,1\}^\lambda$ to be $g(x) = f(x[0:n])$. Is g a OWF?

Sample Problems

Let $f : \{0,1\}^n \to \{0,1\}^\lambda$ be a OWF. Define $g : \{0,1\}^{n+1} \to \{0,1\}^\lambda$ to be $g(x) = f(x[0:n])$. Is g a OWF?
Solution: This is a OWF. Suppose for contradiction that a PPT adversary could invert $y = g(x)$ with non-negligible probability and obtain x' such that $g(x') = y$. Then, the adversary would be able to invert f by simply taking $x'[0:n-1]$. This contradicts the fact that f is a OWF.

Sample Problems

Let $f : \{0,1\}^n \rightarrow \{0,1\}^\lambda$ be a OWF. Let $g : \{0,1\}^{2n} \rightarrow \{0,1\}^\lambda$ where $g(x_1 || x_2) = f(x_1) \oplus x_2$. Does this imply that g is a OWF?

Sample Problems

Let $f : \{0,1\}^n \to \{0,1\}^\lambda$ be a OWF. Let $g : \{0,1\}^{2n} \to \{0,1\}^\lambda$ where $g(x_1||x_2) = f(x_1) \oplus x_2$. Does this imply that g is a OWF?
Solution: No. Suppose we are given $y = g(x_1||x_2)$. Then, let us choose a random x_1' and compute $f(x_1')$. Then, let us choose $x_2' = y \oplus f(x_1')$. We get that $g(x_1'||x_2') = f(x_1') \oplus y \oplus f(x_1') = y$.

Sample Problems

Suppose that $h_1 : \{0,1\}^n \rightarrow \{0,1\}^d$ is a collision resistant hash function. Does it imply that $h_2 : \{0,1\}^{n-d} \times \{0,1\}^n \rightarrow \{0,1\}^d$ is also collision resistant, where $h_2(x,y) = h_1(x || h_1(y))$?

Sample Problems

Suppose that $h_1 : \{0,1\}^n \to \{0,1\}^d$ is a collision resistant hash function. Does it imply that $h_2 : \{0,1\}^{n-d} \times \{0,1\}^n \to \{0,1\}^d$ is also collision resistant, where $h_2(x, y) = h_1(x||h_1(y))$?
Solution: Yes. Suppose h_2 is not collision resistant, so we are able to find x, y, x', y' such that $(x, y) \neq (x', y')$ and $h_2(x, y) = h_2(x', y')$. Therefore, either it is the case that $x||h_1(y) = x'||h_1(y')$ or $x||h_1(y) \neq x'||h_1(y')$. In the first case, that implies that $h_1(y) \neq h_1(y')$, so we have found a collision for h_1. In the second case, $x||h_1(y)$ and $x'||h_1(y')$ cause a collision.

AES

AES is a pseudorandom permutation.

Pseudorandom Function

A function $f : K \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is said to be a pseudorandom function (PRF) if a probabilistic polynomial time adversary A cannot distinguish between given oracle access to $f(k, \cdot)$ for random $k \leftarrow K$ and oracle access to a truly random function $U : \{0,1\}^\lambda \to \{0,1\}^\lambda$:
For all ppt adversaries A, there exists negligible μ such that

$$\left| \Pr\left[A^{f(k,\cdot)}(1^\lambda) = 1 : \ k \xleftarrow{R} K \right] - \Pr\left[A^{U}(1^\lambda) = 1 : \ U \xleftarrow{R} \mathrm{Fun}_{\lambda \to \lambda} \right] \right| \le \mu(\lambda).$$

.

AES

Pseudorandom Permutation

A function $f : K \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is said to be a pseudorandom permutation (PRP) if a probabilistic polynomial time adversary A cannot distinguish between given oracle access to $f(k,\cdot)$ for random $k \leftarrow K$ and oracle access to a truly random permutation $U : \{0,1\}^\lambda \to \{0,1\}^\lambda$, AND f maps distinct inputs to distinct outputs and there exists an efficient inversion algorithm $f^{-1}(k,\cdot)$.

AES

Pseudorandom Permutation

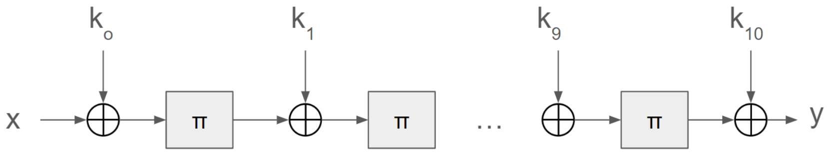
A function $f : K \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ is said to be a pseudorandom permutation (PRP) if a probabilistic polynomial time adversary A cannot distinguish between given oracle access to $f(k,\cdot)$ for random $k \leftarrow K$ and oracle access to a truly random permutation $U : \{0,1\}^\lambda \to \{0,1\}^\lambda$, AND f maps distinct inputs to distinct outputs and there exists an efficient inversion algorithm $f^{-1}(k,\cdot)$.

PRP/PRF Switching Lemma

If the adversary queries for T input/output pairs, then the probability that it can distinguish between a PRP and a PRF is at most $\frac{T(T-1)}{2^{\lambda+1}}$

AES

AES(k, x):



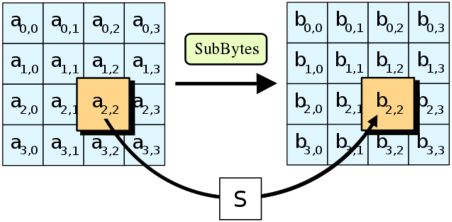
$k_0, \ldots, k_{10}$ are derived from the key k through an invertible algorithm. π is an invertible function consisting of 3 steps: substitute bytes, shift rows, and mix columns (no mix columns in the last round)

AES Steps

AES treats its inputs as a matrix of bytes.

AES Steps

AES treats its inputs as a matrix of bytes.
Substitute bytes: Replaces bytes in the matrix according to a map called the sbox. This adds nonlinearity to the algorithm (the other two steps are linear).

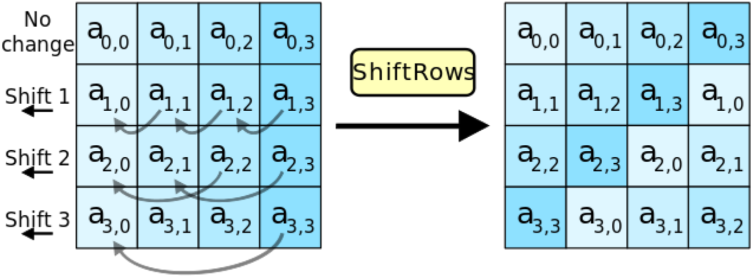


right (low-order) nibble; left (high-order) nibble

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

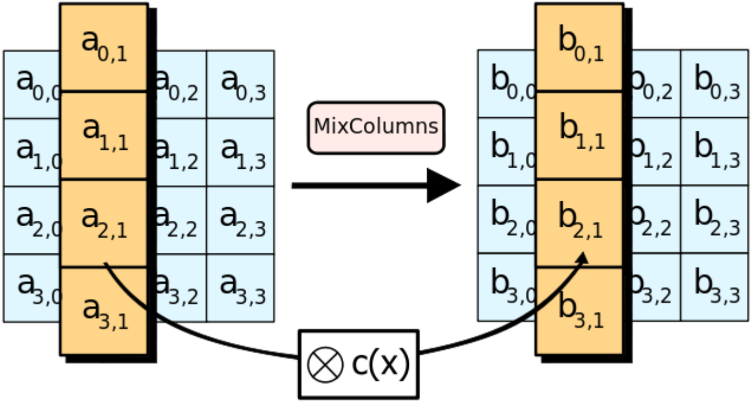
AES Steps

Shift rows: Cyclically shifts the bytes in each row by a certain offset. The first row is left unchanged, the second row is shifted one to the left, the third row is shifted two, and the fourth row is shifted three.



AES Steps

Mix columns: Each column is multiplied with a specific matrix



Linear Algebra Review: Solving $A\mathbf{x} = \mathbf{b}$

Linear Algebra Review: Solving $A\mathbf{x} = \mathbf{b}$

We can solve $A\mathbf{x} = \mathbf{b}$ by using Gaussian elimination to put the matrix in row echelon form.

$$\begin{bmatrix} 1 & 2 & 2 & 2 & b_1 \\ 2 & 4 & 6 & 8 & b_2 \\ 3 & 6 & 8 & 10 & b_3 \end{bmatrix} \to \cdots \to \begin{bmatrix} 1 & 2 & 2 & 2 & b_1 \\ 0 & 0 & 2 & 4 & b_2 - 2b_1 \\ 0 & 0 & 0 & 0 & b_3 - b_2 - b_1 \end{bmatrix}$$

Linear Algebra Review: Solving $A\mathbf{x} = \mathbf{b}$

We can solve $A\mathbf{x} = \mathbf{b}$ by using Gaussian elimination to put the matrix in row echelon form.

$$\begin{bmatrix} 1 & 2 & 2 & 2 & b_1 \\ 2 & 4 & 6 & 8 & b_2 \\ 3 & 6 & 8 & 10 & b_3 \end{bmatrix} \rightarrow \cdots \rightarrow \begin{bmatrix} 1 & 2 & 2 & 2 & b_1 \\ 0 & 0 & 2 & 4 & b_2 - 2b_1 \\ 0 & 0 & 0 & 0 & b_3 - b_2 - b_1 \end{bmatrix}$$

- We can then get a particular solution using back-substitution, setting all free variables to 0: Let $\mathbf{b} = (1, 5, 6)$, then $\mathbf{x}_p = (-2, 0, 3/2, 0)$

Linear Algebra Review: Solving $A\mathbf{x} = \mathbf{b}$

We can solve $A\mathbf{x} = \mathbf{b}$ by using Gaussian elimination to put the matrix in row echelon form.

$$\left[\begin{array}{ccccc} 1 & 2 & 2 & 2 & b_1 \\ 2 & 4 & 6 & 8 & b_2 \\ 3 & 6 & 8 & 10 & b_3 \end{array}\right] \to \cdots \to \left[\begin{array}{ccccc} 1 & 2 & 2 & 2 & b_1 \\ 0 & 0 & 2 & 4 & b_2 - 2b_1 \\ 0 & 0 & 0 & 0 & b_3 - b_2 - b_1 \end{array}\right]$$

- We can then get a particular solution using back-substitution, setting all free variables to 0: Let $\mathbf{b} = (1, 5, 6)$, then $\mathbf{x}_p = (-2, 0, 3/2, 0)$
- To get the complete solution, we add the nullspace (solutions to $A\mathbf{x} = \mathbf{0}$): $\mathbf{x} = (-2, 0, 3/2, 0) + c_1(-2, 1, 0, 0) + c_2(2, 0, -2, 1)$

Linear Algebra Review: Solving $Ax = b$

How many solutions are there?

Linear Algebra Review: Solving $Ax = b$

How many solutions are there?

Definition

Rank: The rank of a matrix is dimension of the vector space spanned by its columns.

Linear Algebra Review: Solving $Ax = b$

How many solutions are there?

Definition

Rank: The rank of a matrix is dimension of the vector space spanned by its columns.

Given a $m \times n$ matrix A, we have the following cases (where R is the reduced row echelon form of A)

	$r = m = n$	$r = n < m$	$r = m < n$	$r < m, r < n$
R	I	$\begin{bmatrix} I \\ 0 \end{bmatrix}$	$\begin{bmatrix} I & F \end{bmatrix}$	$\begin{bmatrix} I & F \\ 0 & 0 \end{bmatrix}$
# solutions to $A\mathbf{x} = \mathbf{b}$	1	0 or 1	infinitely many	0 or infinitely many

Questions?