Blockchain Foundations **NTUA, Spring 2026**

Theory Workshop 1

Problem 1

Let $H^1_\kappa, H^2_\kappa : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$ be families of hash functions. Define $G_{2\kappa}(x) : \{0,1\}^* \longrightarrow \{0,1\}^{2\kappa}$ as $G_{2\kappa}(x) = H^1_\kappa(x)||H^2_\kappa(x)$. Use a reduction to show that $G_{2\kappa}$ is a family of collision-resistant hash functions if at least one of H^1_κ, H^2_κ is collision-resistant.

Proof: For the sake of contradiction, suppose that $G_{2\kappa}$ weren't a family of collision-resistant hash functions. This means that there exists a PPT adversary $\mathcal{A}$ for which

$$\Pr[\mathsf{collision\text{-}game}_{G,\mathcal{A}}(2\kappa) = 1] > \mathrm{negl}(\kappa)$$

We construct adversaries $\mathcal{A}_1, \mathcal{A}_2$ as follows:

Algorithm 1 Adversary $\mathcal{A}_i$ that breaks H^i_κ, $i \in \{1,2\}$.

```
1: function A_i(1^κ)
2:     x_1, x_2 ← A(1^{2κ})
3:     return (x_1, x_2)
4: end function
```

First, we analyze the case for $i = 1$. Since $\mathcal{A}$ is PPT then also $\mathcal{A}_1$ is PPT. Notice that $\mathcal{A}$ wins the collision-resistance game for $G_{2\kappa}$ if $\mathcal{A}_1$ wins the collision-resistance game for H^1_κ. This is because $G_{2\kappa}(x_1) = G_{2\kappa}(x_2) \Leftrightarrow H^1_\kappa(x_1)||H^2_\kappa(x_1) = H^1_\kappa(x_2)||H^2_\kappa(x_2) \Leftrightarrow H^1_\kappa(x_1) = H^1_\kappa(x_2) \wedge H^2_\kappa(x_1) = H^2_\kappa(x_2)$. So if $\mathcal{A}$ finds x_1, x_2 such that $x_1 \neq x_2 \wedge G_{2\kappa}(x_1) = G_{2\kappa}(x_2)$, then $\mathcal{A}_1$ returns the same x_1, x_2 for which $H^1_\kappa(x_1) = H^1_\kappa(x_2)$. Hence,

$$\Pr[\mathsf{collision\text{-}game}_{H^1,\mathcal{A}_1}(\kappa) = 1] \geq \Pr[\mathsf{collision\text{-}game}_{G,\mathcal{A}}(2\kappa) = 1] > \mathrm{negl}(\kappa)$$

Therefore, H^1_κ is not a family of collision-resistants hash functions.
The case for $i = 2$ is porven identically.
This contradicts the assumption that at least one of H^1_κ, H^2_κ is collision-resistant.

Problem 2

It is not sufficient to assume a hash function is collision resistant in order for it to be used in proof-of-work applications. That is why we make use of the random oracle model. To see why, answer the following question: Given a random oracle H (which is collision resistant), define a different collision resistant hash function G which behaves similar to a random oracle for honest parties, but is very beneficial for solving proof-of-work to the adversary. Calculate the probability

that the adversary obtains a block with your hash function G in one query, and the probability that an honest party obtains a block with your hash function in one query.

Recall that in proof-of-work, a miner searches for a value ctr such that $G(s\|\bar{x}\|ctr) < T$, with s and $\bar{x}$ is fixed. To construct a hash function that behaves differently for an adversary than for honest users, we introduce a *trapdoor* in the ctr input. We define G with a distinguished *trapdoor* value $ctr = 0^\kappa$, which is given to the adversary.

First Solution:
Define:

$$G(s||\bar{x}||ctr) = \begin{cases} 0^\kappa||H(s||\bar{x}||ctr) & \text{if } ctr = 0^\kappa \\ H(s||\bar{x}||ctr)||0^\kappa & \text{if } ctr \neq 0^\kappa \end{cases}$$

Claim: G is a collision resistant hash function.
Proof: Let $B = s\|\bar{x}\|ctr$. For every PPT adversary $\mathcal{A}$:

$$\begin{aligned}
&\Pr[\mathsf{collision\text{-}game}_{G,\mathcal{A}}(\kappa) = 1] = \\
&\quad \Pr[G(B_1) = G(B_2) \wedge B_1 \neq B_2 \mid ctr_1 \neq 0^\kappa \wedge ctr_2 \neq 0^\kappa] \cdot \Pr[ctr_1 \neq 0^\kappa \wedge ctr_2 \neq 0^\kappa] \\
&\quad + \Pr[G(B_1) = G(B_2) \wedge B_1 \neq B_2 \mid ctr_1 = ctr_2 = 0^\kappa] \cdot \Pr[ctr_1 = ctr_2 = 0^\kappa] \\
&\quad + \Pr[G(B_1) = G(B_2) \mid (ctr_1 = 0^\kappa \wedge ctr_2 \neq 0^\kappa) \vee (ctr_1 \neq 0^\kappa \wedge ctr_2 = 0^\kappa)] \\
&\quad \cdot \Pr[(ctr_1 = 0^\kappa \wedge ctr_2 \neq 0^\kappa) \vee (ctr_1 \neq 0^\kappa \wedge ctr_2 = 0^\kappa)] \Rightarrow \\
&\Pr[\mathsf{collision\text{-}game}_{G,\mathcal{A}}(\kappa) = 1] \leq \\
&\quad \Pr[G(B_1) = G(B_2) \wedge B_1 \neq B_2 \mid ctr_1 \neq 0^\kappa \wedge ctr_2 \neq 0^\kappa] \\
&\quad + \Pr[G(B_1) = G(B_2) \wedge B_1 \neq B_2 \mid ctr_1 = ctr_2 = 0^\kappa] \\
&\quad + \Pr[G(B_1) = G(B_2) \mid (ctr_1 = 0^\kappa \wedge ctr_2 \neq 0^\kappa) \vee (ctr_1 \neq 0^\kappa \wedge ctr_2 = 0^\kappa)]
\end{aligned}$$

We analyse each term separately:

1. $ctr_1 \neq 0^\kappa \wedge ctr_2 \neq 0^\kappa$: In this case $G(B_i) = H(B_i)\|0^\kappa$, so

   $$G(B_1) = G(B_2) \;\Leftrightarrow\; H(B_1)\|0^\kappa = H(B_2)\|0^\kappa \;\Leftrightarrow\; H(B_1) = H(B_2).$$

   Since H is collision-resistant,

   $$\begin{aligned}&\Pr[G(B_1) = G(B_2) \wedge B_1 \neq B_2 \mid ctr_1 \neq 0^\kappa \wedge ctr_2 \neq 0^\kappa] = \\ &\Pr[\mathsf{collision\text{-}game}_{H,\mathcal{A}}(\kappa) = 1] = \mathrm{negl}(\kappa).\end{aligned}$$

2. $ctr_1 = ctr_2 = 0^\kappa$: Same as above.

3. $(ctr_1 = 0^\kappa \wedge ctr_2 \neq 0^\kappa) \vee (ctr_1 \neq 0^\kappa \wedge ctr_2 = 0^\kappa)$: Without loss of generality let $ctr_1 = 0^\kappa$ and $ctr_2 \neq 0^\kappa$. Then $G(B_1) = 0^\kappa\|H(B_1)$ and $G(B_2) = H(B_2)\|0^\kappa$. For these to be equal we need $H(B_1) = 0^\kappa$ and $H(B_2) = 0^\kappa$. So $H(B_1) = H(B_2)$ and $B_1 \neq B_2$ so this is a collision, and since H is collision resistant:

   $$\Pr[G(B_1) = G(B_2) \wedge B_1 \neq B_2 \mid ctr_1 \oplus ctr_2 \neq 0^\kappa] = \mathrm{negl}(\kappa).$$

Therefore, all three terms are negligible, and thus $\Pr[\mathsf{collision\text{-}game}_{G,\mathcal{A}}(\kappa) = 1]$ is negligible. Hence, G is collision-resistant.
The adversary who knows the *trapdoor* value acts as follows: She chooses the $s, \bar{x}$ part of the block B and adds the *trapdoor* value for the ctr. So, she constructs the block $B = (s, \bar{x}, 0^\kappa)$. On the other hand, the honest parties mine using the usual mining algorithm.
Assume that $T \geq 2^\kappa$. Then the probability that the adversary obtains a block is

$$\Pr[G(B) < T \mid ctr = 0^\kappa] = 1.$$

For an honest party, in order to find a block it can either guess the *trapdoor* value or find a value B such that $G(B) = H(B)\|0^\kappa = H(B) \cdot 2^\kappa < T$. So the total probability that an honest party obtains a block is

$$\Pr[G(B) < T \mid ctr \neq 0^\kappa] \cdot \Pr[ctr \neq 0^\kappa] + \Pr[G(B) < T \mid ctr = 0^\kappa] \cdot \Pr[ctr = 0^\kappa] =$$
$$\frac{T}{2^{2\kappa}} \cdot \frac{2^\kappa - 1}{2^\kappa} + 1 \cdot \frac{1}{2^\kappa} = \frac{T(2^\kappa - 1) + 2^{2\kappa}}{2^{3\kappa}} \simeq \frac{T + 2^\kappa}{2^{2\kappa}} \simeq \frac{T}{2^{2\kappa}},$$

which would be the behavior of a random oracle with 2κ output bits.
If $T < 2^\kappa$, then the probability that the adversary obtains a block is

$$\Pr[G(B) < T \mid ctr = 0^\kappa] = \frac{T}{2^\kappa}.$$

For an honest party, in order to find a block it can either guess the *trapdoor* value together with a value B such that $H(B) < T$ or find a value B such that $G(B) = H(B)\|0^\kappa = H(B) \cdot 2^\kappa < T$. Since $T < 2^\kappa$ this hold only if $H(B) = 0^\kappa$. So the total probability that an honest party obtains a block is

$$\Pr[G(B) < T \mid ctr \neq 0^\kappa] \cdot \Pr[ctr \neq 0^\kappa] + \Pr[G(B) < T \mid ctr = 0^\kappa] \cdot \Pr[ctr = 0^\kappa] =$$
$$\frac{1}{2^\kappa} \cdot \frac{2^\kappa - 1}{2^\kappa} + \frac{T}{2^\kappa} \cdot \frac{1}{2^\kappa} = \frac{1}{2^\kappa} \cdot \frac{T + 2^\kappa - 1}{2^\kappa} = \mathrm{negl}(\kappa).$$

Second Solution:
Define:

$$G(s||\bar{x}||ctr) = \begin{cases} 0^\kappa||H(s||\bar{x}||ctr) & \text{if } ctr = 0^\kappa \\ H(s||\bar{x}||ctr)||H(s||\bar{x}||ctr) & \text{if } ctr \neq 0^\kappa \end{cases}$$

G is collision-resistant (similar argurment as in the first solution).

Let $B = s\|\bar{x}\|ctr$.
Probability Calculation: The adversary who knows the *trapdoor* value acts the same as in the *first solution*, and the probability of the adversary to find a block remains the same. For an honest party, in order to find a block it can either guess the *trapdoor* value together with a value B such that $H(B) < T$ or find a value B such that $G(B) = H(B)||H(B) = H(B) \cdot 2^\kappa + H(B) < T \Rightarrow H(B) < \frac{T}{2^\kappa + 1}$. So the total probability that an honest party obtains a block is:

$$\Pr[G(B) < T \mid ctr \neq 0^\kappa] \cdot \Pr[ctr \neq 0^\kappa] + \Pr[G(B) < T \mid ctr = 0^\kappa] \cdot \Pr[ctr = 0^\kappa] =$$
$$\Pr\left[H(B) < \frac{T}{2^\kappa + 1}\right] \cdot \frac{2^\kappa - 1}{2^\kappa} + \Pr\left[H(B) < T\right] \cdot \frac{1}{2^\kappa}$$

- If $T > 2^\kappa$:

  $$\Pr\left[H(B) < \frac{T}{2^\kappa + 1}\right] = \frac{T}{2^\kappa(2^\kappa + 1)},$$
  $$\Pr\left[H(B) < T\right] = 1$$

  So, the total probability that an honest party obtains a block is:

  $$\frac{T}{2^\kappa(2^\kappa + 1)} \cdot \frac{2^\kappa - 1}{2^\kappa} + 1 \cdot \frac{1}{2^\kappa} \simeq \frac{T}{2^{2\kappa}}$$

  which would be the behavior of a random oracle with 2κ output bits.

- If $T \leq 2^\kappa$:

  $$\Pr\left[H(B) < \frac{T}{2^\kappa + 1}\right] = \frac{1}{2^\kappa},$$

  because the only value that satisfies this inequality is $H(B) = 0^\kappa$, and

  $$\Pr\left[H(B) < T\right] = \frac{T}{2^\kappa}$$

  So, the total probability that an honest party obtains a block is:

  $$\frac{1}{2^\kappa} \cdot \frac{2^\kappa - 1}{2^\kappa} + \frac{T}{2^\kappa} \cdot \frac{1}{2^\kappa} \leq \frac{1}{2^\kappa} \cdot \frac{2^\kappa - 1}{2^\kappa} + 1 \cdot \frac{1}{2^\kappa} = negl(\kappa)$$

Problem 3

Consider the complete transaction graph ***accepted and stored in the database of an honest node*** illustrated in Figure 1. The circles represent transactions and the numbers within them are the txids. The outgoing edges are outputs, and the incoming edges are inputs. For each transaction, edges appear ordered from top to bottom. Assume a system in which coinbase transactions are allowed to have multiple outputs whose total value does not add up to more than the macroeconomic policy. For this graph:

1. Identify the coinbase transactions.
2. Identify the double spending transactions. For each double spending transaction, identify the transaction it is conflicting with.
3. Identify the UTXO set.
4. Identify the transactions for which the Weak Conservation Law does not hold.
5. Identify the outputs of transactions that are partially, but not fully, spent.

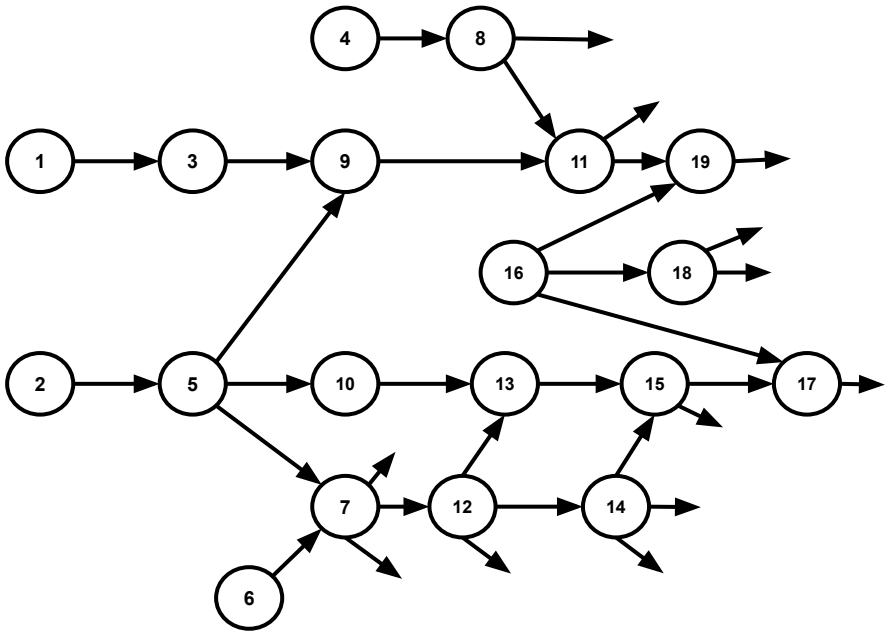

Figure 1: The transaction graph of Problem 4.

To identify transactions, use their txids. To identify an output, use outpoint notation. Note that transaction 6 has four outputs.

1. Coinbase transactions are transactions with no inputs and one or more outputs. They correspond to transactions $1, 2, 4, 6, 16$ in the graph.
2. Honest nodes won't accept double spends, so there are no double spends in the graph.
3. The UTXO set consists of edges that don't point to other transactions. In the transaction graph, this corresponds to the following outpoint set:

   $\{(7,0), (7,2), (8,0), (11,0), (12,2), (14,1), (14,2), (15,1), (17,0), (18,0), (18,1), (19,0)\}$
4. Honest nodes will only accept non-coinbase transactions that pass the Weak Conservation Law, so only coinbase transactions will violate it. Hence, we will have the same answer as in part 1 (i.e., transactions $1, 2, 4, 6, 16$).
5. It is not possible to only spend part of a transaction's output. They are either not spent (part of UTXO set) or completely spent.

Problem 4

Assume there exists an adversary $\mathcal{A}$ that breaks unforgeability with probability

$$\Pr[\mathsf{existential\text{-}forgery\text{-}game}_{(\mathsf{Gen},\mathsf{Sig},\mathsf{Ver}),\mathcal{A}}(\kappa) = 1] = p$$

We construct an adversary $\mathcal{A}'$ that runs $\mathcal{A}$ independently $q(\kappa)$ times, where $q(\kappa)$ is a polynomial in the security parameter κ, and outputs success if at least one execution of $\mathcal{A}$ succeeds. Find the probability of success of the adversary $\mathcal{A}'$.

Let E_i denote the event that the i-th execution of $\mathcal{A}$ succeeds, for $i = 1, \ldots, q(\kappa)$, so

$$\Pr[E_i] = p$$

Since $\mathcal{A}'$ succeeds if and only if at least one execution succeeds,

$$\Pr\left[\mathsf{existential\text{-}forgery\text{-}game}_{(\mathsf{Gen},\mathsf{Sig},\mathsf{Ver}),\mathcal{A}'}(\kappa) = 1\right] = \Pr\left[\bigcup_{i=1}^{q(\kappa)} E_i\right].$$

From the union bound we can bound this probability:

$$\Pr\left[\bigcup_{i=1}^{q(\kappa)} E_i\right] \leq \sum_{i=1}^{q(\kappa)} \Pr[E_i] = q(\kappa)\, p.$$

Hence,

$$\Pr\left[\mathsf{existential\text{-}forgery\text{-}game}_{(\mathsf{Gen},\mathsf{Sig},\mathsf{Ver}),\mathcal{A}'}(\kappa) = 1\right] \leq q(\kappa)\, p.$$

Notice that, if $p = \mathrm{negl}(\kappa)$ then the bound is also negligible, since $\mathrm{poly}(\kappa) \cdot \mathrm{negl}(\kappa) = \mathrm{negl}(\kappa)$.

Exact probability. Since the $q(\kappa)$ executions are independent and each of these fail with probability $(1-p)$, the probability that all of them fail is $(1-p)^{q(\kappa)}$. So the probability of at least one of them succeeds:

$$\Pr\left[\mathsf{existential\text{-}forgery\text{-}game}_{(\mathsf{Gen},\mathsf{Sig},\mathsf{Ver}),\mathcal{A}'}(\kappa) = 1\right] = 1 - (1-p)^{q(\kappa)}.$$

Reference

Some helpful definitions are provided below. For the full definitions, consult the lecture notes.

Definition (Collision Resistance)**.** A hash function $H : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$ is *collision resistant* if for all PPT adversaries $\mathcal{A}$,

$$\Pr[\mathsf{collision\text{-}game}_{H,\mathcal{A}}(\kappa) = 1] = \mathrm{negl}(\kappa)\,.$$

The game is defined in Algorithm 2.

Algorithm 2 The collision-finding game for a hash function H.

function COLLISION-GAME$_{H,\mathcal{A}}(\kappa)$
  $x_1, x_2 \leftarrow \mathcal{A}(1^\kappa)$
  return $H_\kappa(x_1) = H_\kappa(x_2) \wedge x_1 \neq x_2$
end function

Definition (Correct Signature)**.** A signature scheme $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is *correct* if, for all $m \in \{0,1\}^*$, whenever $(sk, pk) \leftarrow \mathsf{Gen}(1^\kappa)$, we have that $\mathsf{Ver}(pk, m, \mathsf{Sig}(sk, m)) = 1$.

Definition (Secure Signature)**.** A signature scheme $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is *secure* if for all PPT adversaries $\mathcal{A}$,

$$\Pr[\mathsf{existential\text{-}forgery\text{-}game}_{(\mathsf{Gen},\mathsf{Sig},\mathsf{Ver}),\mathcal{A}}(\kappa) = 1] = \mathrm{negl}(\kappa)\,.$$

The game is defined in Algorithm 3.

Algorithm 3 The existential forgery game for a signature scheme (Gen, Sig, Ver).

function EXISTENTIAL-FORGERY-GAME$_{(\mathrm{GEN},\mathrm{SIG},\mathrm{VER}),\mathcal{A}}(\kappa)$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
  $M \leftarrow \emptyset$
  function $\mathcal{O}$(m)
    $M \leftarrow M \cup \{m\}$
    return $\mathsf{Sig}(sk, m)$
  end function
  $m, \sigma \leftarrow \mathcal{A}^{\mathcal{O}}(pk)$
  return $\mathsf{Ver}(pk, \sigma, m) \wedge m \notin M$
end function

Algorithm 4 The proof-of-work algorithm.

function PoW$_{H,T}$
  $\mathsf{ctr} \xleftarrow{\$} \{0,1\}^\kappa$
  while true **do**
    $B \leftarrow \mathsf{ctr}$
    if $H(B) \leq T$ **then**
      return B
    end if
    $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$
  end while
end function