Spring 2023

MATH 1025: Introduction to Cryptography

Take-Home Exam

Zero-knowledge proofs

Problem 1. (a) [5 **pts**] Find the solutions to the congruence $x^2 \equiv 4 \pmod{77}$.[1]

(b) [5 **pts**] Use your answer in (a) to find the prime factorization of 77 (show that $77 = 7 \cdot 11$).[2]

In cryptography, a proof of knowledge is an interactive proof in which the prover succeeds in 'convincing' a verifier that the prover knows something.

Definition. A **zero-knowledge proof (protocol)** is a proof that satisfies three properties:

(1) **Completeness**: if the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.

(2) **Soundness**: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.

[1] **Hint:** Notice that $77 = 7 \cdot 11$ and separately solve the congruences $x^2 \equiv 4 \pmod{7}$ and $x^2 \equiv 4 \pmod{11}$ (you can get the solutions by simply plugging in all possible residues). You will get two solutions for both congruences ((a_1, a_2) and (b_1, b_2)). Then, for each pair (a_i, b_j) construct a number c_{ij} with $c_{ij} \equiv a_i \pmod{7}$ and $c_{ij} \equiv b_j \pmod{11}$, using Chinese remainder theorem. This way you will get the four solutions of the initial congruence.

[2] **Hint:** Notice that $c_{11} - c_{12} \equiv 0 \pmod{7}$ and $c_{12} - c_{22} \equiv 0 \pmod{11}$. This implies that you can obtain the prime factors of 77 as $\gcd(77, c_{11} - c_{12})$ and $\gcd(77, c_{12} - c_{22})$.

(3) **Zero-knowledge**: if the statement is true, no verifier learns anything other than the fact that the statement is true. In other words, just knowing the statement (not the secret) is sufficient to imagine a scenario showing that the prover knows the secret. This is formalized by showing that every verifier has some simulator that, given only the statement to be proved (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the verifier in question.

Problem 2. Let n be an integer which is a product of two distinct odd primes. It is also known that the primes are equivalent modulo 4. Peggy was proposed the following protocol in order to convince Victor that she knows the prime factors of n, i.e. p and q with $n = pq$.

Step 1. Victor, who knows n, but not p and q, chooses a number $1 < x < n$ and sends $y = x^4 \pmod{n}$, $0 < y < n$ to Peggy.

Step 2. When Peggy receives y she computes the square roots of y modulo n (there are four, see Problem 1) and sends the root z with $\left(\frac{z}{p}\right) = \left(\frac{z}{q}\right) = 1$ back to Victor (again, $0 < z < n$).

Step 3. Victor receives z checks that $z \equiv x^2 \pmod{n}$ and is convinced that Peggy can take square roots modulo n, which implies that she knows the prime factors of n. [3]

(a) [10 **pts**] Show that Victor can eventually (by testing Peggy's claim according to the protocol (possibly) multiple times) find out the factorization of n and, hence, the protocol does not have the zero-knowledge property.[4]

(b) [5 **pts**] Where did you use the assumption that p and q are equivalent modulo 4?

[3] The problem of finding square roots modulo n is known to be equivalent to integer factorization of n, there is a fast algorithm for finding square roots modulo prime numbers (see the end of Chapter 2.2 in [Kob94] for details).

[4] **Hint:** At some point Victor will pick a number x with $\left(\frac{x}{p}\right) = 1$, $\left(\frac{x}{q}\right) = -1$ or $\left(\frac{x}{p}\right) = -1$ $\left(\frac{x}{q}\right) = 1$ (this happens with probability 50%). On the other hand, Peggy will return a number z with $\left(\frac{z}{p}\right) = \left(\frac{z}{q}\right) = 1$.

(a) Show that $z \neq \pm x^2$ (use Jacobi symbols).

(b) Generalize the algorithm you used in Problem 1(b) to show how Victor can find a prime factor of n.

Schnorr protocol

One of the simplest and frequently used proofs of knowledge, the proof of knowledge of a discrete logarithm, is due to Schnorr. Let $G = \langle g \rangle$ be a cyclic group of order N generated by g. In order to prove the knowledge of $x = \log_g y$, the prover interacts with the verifier as follows:

Step 1. Samantha randomly chooses a number $1 < r < N$ and sends the message $t = g^r$ to the verifier Victor.

Step 2. Victor replies with a challenge number $\beta \in \{0, 1\}$ chosen at random (e.g. determined by an outcome of a coin flip).

Step 3. After receiving β, Samantha sends the third and last message (the response) $s \equiv r + \beta x \pmod{N}$.

The verifier accepts, if $g^s = t y^\beta$.[5]

Problem 3. (a) [2 **pts**] Check that the procedure actually works, i.e. $g^s = t y^\beta$ in case the incoming data was generated according to the protocol.

(b) [8 **pts**] Suppose that Samantha does not actually know $x = \log_g y$ and sends either $t = g^r$ and $s = r$ or $t = g^r y^{-1}$ and $s = r$ on steps 1 and 3. What are the odds (probability) that she can fool Victor 10 times in a row?

[5] We sketch the argument on why the protocol has zero-knowledge property. It is enough to show that a person, who does not know $x = \log_g y$ but does know in advance the sequence of values of β, can simulate the same response as Samantha (by sending $t = g^r$ and $s = r$ for $\beta = 0$; $t = g^r y^{-1}$ and $s = r$ in case $\beta = 1$ on steps 1 and 3).

A few words on discriminants

Problem 4. (a) Let $f(x) = x^2 + bx + c$ be a quadratic polynomial.

(1) [2 **pts**] Denote the zeros of $f(x)$ by x_1 and x_2. Show that $b = -(x_1 + x_2)$ and $c = x_1x_2$.[6]

(2) [3 **pts**] Deduce that $D_f = (x_1 - x_2)^2 = b^2 - 4c$.

(b) Let $f(\widetilde{x}) = \widetilde{x}^3 + \alpha\widetilde{x}^2 + \beta\widetilde{x} + \gamma$ be a cubic polynomial.

(1) [2 **pts**] Find the substitution $x = \widetilde{x} - c$ for which $f(x) = x^3 - ax + b$.

(2) [3 **pts**] Denote the zeros of $f(x)$ by x_1, x_2 and x_3. Show that

$$\begin{cases} x_1 + x_2 + x_3 = 0 \\ a = -(x_1x_2 + x_1x_3 + x_2x_3) \\ b = -x_1x_2x_3. \end{cases}$$ [7]

[6] **Hint:** by Bezout's theorem $f(x) = (x - x_1)(x - x_2)$.
[7] **Hint:** by Bezout's theorem $f(x) = (x - x_1)(x - x_2)(x - x_3)$.

(3) [4 **pts**] Recall (see the proposition on pages 2 - 3 of 'Lectures 19 - 20' notes) that $f(x)$ has multiple zeros ($D_f = 0$) iff $f(x)$ and $f'(x)$ have a common zero, i.e. there exists an x with $f(x) = f'(x) = 0$. Find the x-coordinates (x_α and x_β) of the zeros of $f'(x)$ (express them in terms of a and b).[8]

(4) [4 **pts**] The equations $f(x_\alpha) = 0$ and $f(x_\beta) = 0$ will give rise to (very similar) relation on a and b. Show (by easy algebraic manipulations) that either of these relations gives rise to $4a^3 - 27b^2$.

(5) [2 **pts**] Show (using (b)(2)) that the polynomials $D_f = (x_1 - x_2)^2(x_1 - x_3)^2(x_2 - x_3)^2$ and $4a^3 - 27b^2$ have the same degree. It is not hard to see that they both vanish with degree 2 on the hyperplanes $x_i = x_j$ (you can skip that part). It follows that $D_f = \gamma(4a^3 - 27b^2)$, where $\gamma \neq 0$ is a constant. Check that $\gamma = 1$, which allows to conclude that $D_f = 4a^3 - 27b^2$.

[8] Nobody said the expression must involve both a and b.

Group structure on elliptic curve

Let $\mathbb{P}^2$ be the set of all one-dimensional subspaces (lines through the origin) in a three-dimensional vector space. The points on $\mathbb{P}^2$ are defined by three coordinates up to simultaneous rescaling and denoted by $p = [x : y : z]$. As $[x : y : z] \sim [tx : ty : tz]$ give rise to the same point in $\mathbb{P}^2$ (define the same line through the origin) for any $t \neq 0$, it only makes sense to work with homogeneous polynomials (each monomial of the same degree) in x, y and z. Let $E : \{[x : y : z] \in \mathbb{P}^2 \mid y^2z = x^3 + axz^2 + bz^3\} \subset \mathbb{P}^2$ be an elliptic curve.

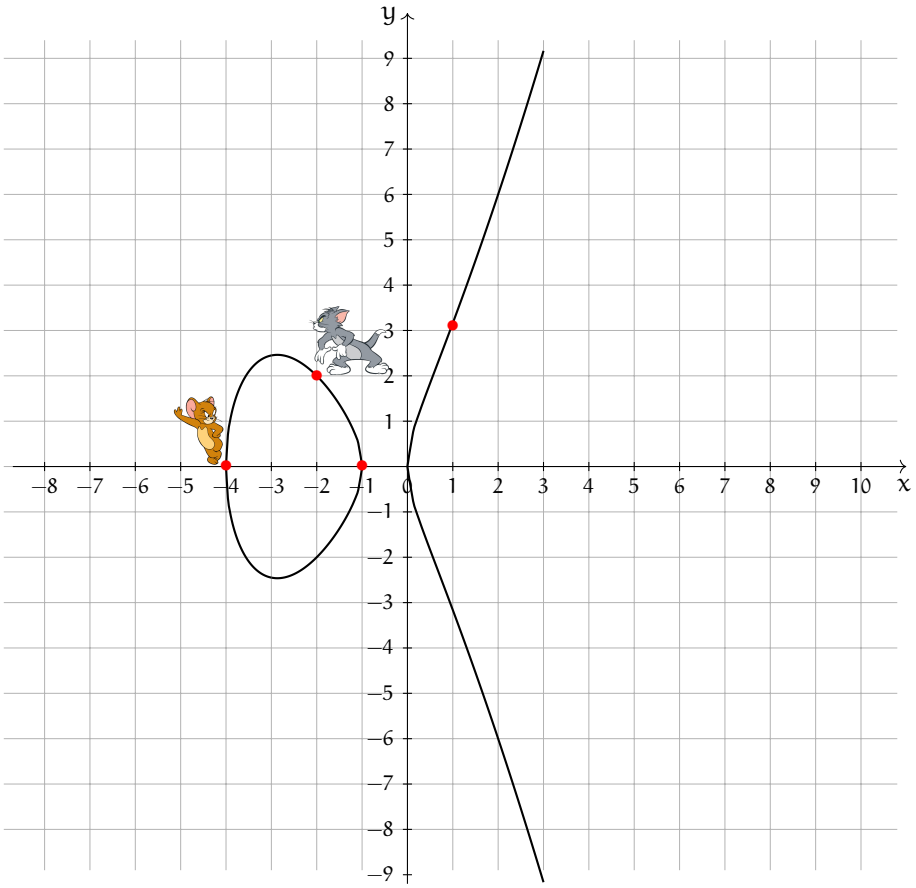
Remark. In class (see 'Lectures 19 - 20' notes) we 'looked' at the elliptic curve away from the line $\{z = 0\} \subset \mathbb{P}^2$, i.e. on the open subset $U_{z\neq 0} = \mathbb{P}^2 \setminus \{z = 0\}$. As each point $[x : y : z] \in U_{z\neq 0}$ is equivalent to $\frac{1}{z}[x : y : z] = [\frac{x}{z} : \frac{y}{z} : 1]$, the defining equation of E becomes $y^2 = x^3 + ax + b$ (we simply put $z = 1$).

Remark. The set $\mathbb{P}^2$ is called the **projective plane**. Analogously one can define projective spaces of any dimension.

Next consider the set of **finite** expressions (formal sums) $\mathcal{P} := \{\sum_{P\in E} n_P P \mid n_p \in \mathbb{Z}\}$ with a free abelian group structure.

Definition. The formal sums $D = \sum_{P\in E} n_p P \in \mathcal{P}$ as above are called **divisors**. Let $D = \sum_{P\in E} n_p P \in \mathcal{P}$ be a divisor, the **degree** of D is the integer $\deg(D) = \sum n_p$.

Example. Let P, Q, R, S be some points on E and consider the divisors $D_1 = 2P - 3Q + 4S$ and $D_2 = P + R - 3S$. Then the divisor $D_3 = 2D_2 - D_1$ is $D_3 = 2D_2 - D_1 = 2P + 2R - 6S - (2P - 3Q + 4S) = 2R - 10S + 3Q$ and the degree of D_1 is $\deg(D_1) = 2 - 3 + 4 = 3$.



Problem 5. We will work with the elliptic curve $E : Y^2 = X(X + 1)(X + 4)$. Let  $= (-4, 0)$ and  $= (-2, 2)$ be two points on E.

(a) [5 **pts**] Consider the divisors $D_1 = 3$  $- 5$  $+ 3(-1, 0)$ and $D_2 = 2$  $+$  $- 2(1, \sqrt{10})$ and find

(1) $D_1 + 2D_2 =$

(2) $3D_1 - D_2 =$

(b) [5 **pts**]

(1) $\deg(D_1) =$

(2) $\deg(D_2) =$

(3) $\deg(D_1 + 2D_2) =$

(4) $\deg(3D_1 - D_2) =$

(c) [5 **pts**] Show that in general for any two divisors $D, D' \in \mathcal{P}$ one has $\deg(D + D') = \deg(D) + \deg(D')$. In other words, the map

$$\deg : \mathcal{P} \to \mathbb{Z}$$

is a group homomorphism.

We will work with the subset $\mathcal{P}^0 \subset \mathcal{P}$, which consists of degree 0 elements.

Remark. Notice that $\mathcal{P}^0$ is the kernel of the homomorphism deg, hence, a subgroup of $\mathcal{P}$.

Let $\sim$ be an equivalence relation on $\mathcal{P}^0$ generated by

$$P_1 + P_2 + P_3 \sim Q_1 + Q_2 + Q_3$$

iff $P_1, P_2, P_3 \in \ell_1$ and $Q_1, Q_2, Q_3 \in \ell_2$ for some lines ℓ_1 and ℓ_2.

Let $\mathcal{O}$ be the point $[0:1:0]$.

Remark. This is the 'mysterious' point that we did not explicitly define in class, since it is 'hidden' on the line $\{z = 0\} \subset \mathbb{P}^2$, which we did not 'see' on $U_{z\neq 0}$.

Problem 6. [5 **pts**] Show that the line $z = 0$ intersects E only at $\mathcal{O}$, but with multiplicity 3. [9]

Problem 7. Let $D = \sum_{P\in E} n_P P \in \mathcal{P}^0$.

(a) [5 **pts**] Show that $D \sim \widetilde{D} = \sum_{Q\in E} n_Q Q - m\mathcal{O}$ with $n_Q \in \mathbb{Z}_{>0}$ and $m = -\sum n_Q$.[10]

(b) [10 **pts**] Show by induction on $n = \sum n_Q$ that $\widetilde{D} \sim P - \mathcal{O}$. [11]

Remark. Let G_E be the group $\mathcal{P}^0/_\sim$. We have established a surjection of sets

$$\varphi : E \to G_E, \varphi(P) = P - \mathcal{O}.$$

It can be shown that φ is one-to-one[12] and, thus an isomorphism. Therefore the elliptic curve has a group structure G_E.

[9] **Hint:** let $f(x)$ be the restriction of the defining equation of E to the line $z = 0$ and check that $f(0) = f'(0) = f''(0) = 0$.

[10] **Hint:** if $n_P < 0$, consider the line ℓ through the points P and $R = \ominus P$, then $P + R + \mathcal{O} \sim 3\mathcal{O}$...

[11] **Hint:** for the induction step, draw a line ℓ through two points Q_1 and Q_2 with nonzero coefficients in $\widetilde{D}$ (or a tangent line to a point Q with $n_Q \geq 2$) and use that $Q_1 + Q_2 + R \sim R + (\ominus R) + \mathcal{O}$ (or $2Q + R \sim R + (\ominus R) + \mathcal{O}$), where R is the third point in $E \cap \ell$.

[12] Not so hard to show, but requires a bit of knowledge in Algebraic Geometry, so we will skip that part.

Pollard's algorithm

The description of the algorithm can be found on page 7 of 'Lectures 17 - 18' notes.

Problem 8. Consider the elliptic curve $E : Y^2 = X^3 - 7X + 13$ over $\mathbb{F}_{137}$. Let $P = [4, 7]$, $Q = [38, 97]$, the order of P is $N = 138$ (it is a generator). Our next goal is to solve the DLP, for P and Q, that is, find a positive integer s such that $Q = sP$. We will use Pollard's algorithm. Take the set $S = \{\text{points on } E\}$ and partition it into three subsets: $S = S_1 \sqcup S_2 \sqcup S_3$ with $S_1 = \{Z \in E \mid 0 \leq x(Z) < 46\}$, $S_2 = \{Z \in E \mid 46 \leq x(Z) < 92\}$ and $S_3 = \{Z \in E \mid 92 \leq x(Z) < 137\}$. Consider the function $f : S \to S$ given by

$$f(Z) = \begin{cases} Z \oplus P, & Z \in S_1 \\ 2Z, & Z \in S_2 \\ Z \oplus Q, & Z \in S_3. \end{cases}$$

(a) [5 **pts**] Let the starting point be $x_1 = \mathcal{O}$. Compute the first 40 elements of the sequence $\{x_1, x_2, \ldots\}$ with $x_n = f^{\circ n}(x_1)$. Denote $\mathcal{X}_{40} := \{x_1, x_2, \ldots, x_{40}\}$.

(b) [5 **pts**] Using your result in (b), derive the first 20 elements of the sequence $\{y_1, y_2, \ldots, y_{20}\}$ with $y_n = x_{2n} = f^{\circ 2n}(x_1)$. Denote $\mathcal{Y}_{20} := \{y_1, y_2, \ldots, y_{20}\}$.

(c) [5 **pts**] Pick an integer $1 \leq i \leq 20$, s.t. $x_i = y_i =: L$. Using that $L = a_i P + b_i Q = c_i P + d_i Q$, write $(a_i - c_i)P = (d_i - b_i)Q = (d_i - b_i)\log_P Q \cdot P$, which implies the congruence $(d_i - b_i)s \equiv (d_i - b_i)\log_P Q \equiv a_i - c_i \pmod{138}$. The numbers $d_i - b_i$ and 138 will not be coprime ($\gcd(d_i - b_i, 138) > 1$), implying $d_i - b_i$ is not invertible modulo 138. Solve the congruence $(d_i - b_i)\log_P Q \equiv a_i - c_i \pmod{23}$ and using that $138 = 23 \cdot 6$ and CRT, list the possible values of s.

(d) [5 **pts**] Choose the correct value of s (trial and error).

Hash functions

Definition. A **hash function** is any function that can be used to map data of arbitrary size to fixed-size values (can be encoded by ℓ bits, in other words, the range of a hash function is $\{0,1\}^{\ell}$).

Below we list some important properties of hash functions.

(1) **One way (OW)** (preimage resistance) Given $y \in \{0,1\}^{\ell}$, it is infeasible to find x with $h(x) = y$.

(2) **Collision resistance (CR)** It is infeasible to find $x \neq x'$ with $h(x) = h(x')$.

(3) **Weak (target) collision resistance (TCR)** Given x, it is infeasible to find $x' \neq x$ with $h(x') = h(x)$.

(4) **Pseudo-randomness (PRF)** Indistinguishable from a truly random function of the input.

$\star$ **Problem 9.** (a) [7 **pts**] Construct an example of a hash function h that is OW but not TCR.

(b) [8 **pts**] Construct an example of a hash function h that is TCR but not OW.

Suppose we have a CR function h on the domain of short messages and would like to use it to construct a CR function H on the domain of long(er) messages. The following commonly used construction is due to Damgard and Merkle.

Step 1. Subdivide your message m into a collection of messages $m = m_0||m_1||\dots||m_n$, each of which has **maximal** length 'digestible' by h (with the possible exception of m_n).

Step 2. Find out the initial value (initialization vector) $\mathtt{Init.value}$.

Step 3. Hash $\mathtt{Init.value}||m_0$ by h, giving rise to $h(\mathtt{Init.value}||m_0)$, then hash this value concatenated with m_1, getting $h(h(\mathtt{Init.value}||m_0)||m_1)$, proceed by taking the hash of the outcome with the next portion of m on each iteration. On the last step hash the result of the preceding one with the final part of the message and padding block ($m_n||\mathrm{PB}$), where $\mathrm{PB} = 100\dots0||\ell en(m)$.

Remark. In case the length $m_n||\mathrm{PB}$ exceeds the maximal size to be hashed by h, we simply add another block in the subdivision, i.e. $m = m_0||m_1||\dots m'_n||m''_n$ with $m = m'_n||m''_n$ and $m''_n||\mathrm{PB}$ having appropriate length.

The procedure is summarized in Figure 1 below.

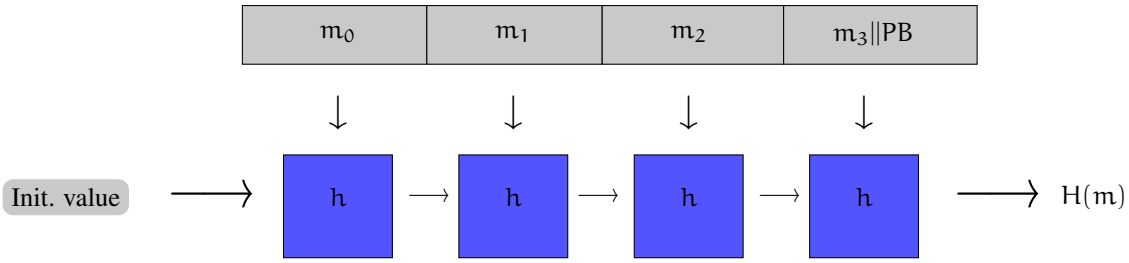


Figure 1: Merkle-Damgard iterated construction (schematically)

Problem 10. [10 **pts**] Show that if h is collision resistant, then so is H. [13]

Merkle trees

A typical block in a bitcoin blockchain contains $N \geq 2000$ transactions. Suppose we want to check if the suggested block is valid. Verifying the (hashes of) all transactions in the block one by one is not practical (requires N verifications). Instead the transactions are organized into a binary tree. Such structure allows to minimize the number of verifications to approximately $\log_2(N)$, which is a very significant reduction.

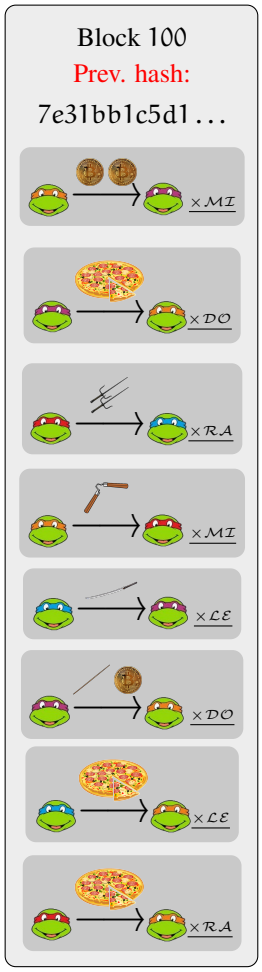
Definition. A **hash tree** or **Merkle tree** is a tree in which every leaf node is labelled with the cryptographic hash of a data block, and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes.

Example. Consider the four friends: Leonardo, Michelangelo, Raphael and Donatello. They love practising martial arts and eating pizza, so they exchange their weapons (katana, nunchaku, sai and bo), share pizza and sometimes pay with bitcoins. All exchanges (transactions) are hashed and recorded on blocks, which are put together in line to form a 'Ninja' blockchain. In addition, each block contains its order in the blockchain and the hash of the preceding block (encoding all info of that block). A typical block is represented on Figure 2 below.

The corresponding binary Merkle tree can be seen on Figure 3.

Definition. Let A and B be two vertices on a graph. The **distance** between A and B is the length of the shortest nonoriented path connecting the vertices. We will denote it by $d(A, B)$.

[13] **Hint.** Argue by contradiction: assume that H is not CR, then there exist $m \neq m'$ with $H(m) = H(m')$. Carefully unravel what that means and show that in this case h is not CR either (notice that the subdivisions $m = m_0||m_1||\ldots||m_r$ and $m' = m'_0||m'_1||\ldots||m'_s$ may contain different number of parts).



122e427e8ac48baf55169d18082f4b4dad6d452456d1b07d5d0e19bafd281829

e5a1efbe5f5ea45d916233dbbe8a8666fb64b0eae8d4b9a2d243aa2e18a1ec97

1314ac57aad0b95b159a074a9fd4fd9a2fa4784d7472afd42f961e83c6e53109

a9f19dcfe87867bc8a9b349307ac23169f2c3263456b5ac4dad0e7dcc4d4ed66

da41262bb41bcf13d116c6d4d09470eef6eca382551f2620c3501dd85099bc39

441a072a1fcf38c5eb58fbf7ebcf88b4805c76f70cd8aa0a90cc75223d6e0ac5

134d530bd05e072425a72ae677555c394bef2352231c6046c6748c3091f1ab45

0f7a0d8500d992ecdaea2279bc94130e634bb1c1c4d74f30879a46b7d8ff0a81

Figure 2: A block in Ninja blockchain

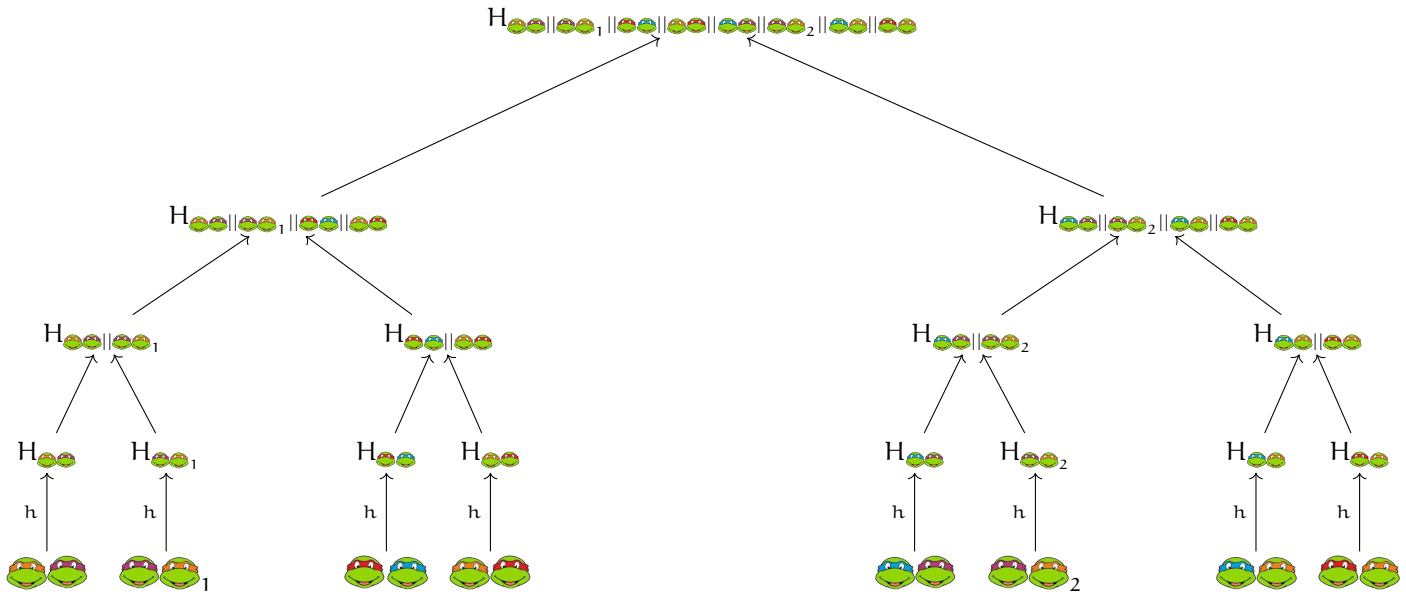


Figure 3: Binary Merkle tree for a block in Ninja blockchain

Example. Consider the binary Merkle tree on Figure 3.

$$d(H_{\bullet\bullet}, H_{\bullet\bullet_1}) = 2$$
$$d(H_{\bullet\bullet}, H_{\bullet\bullet_2}) = 6.$$

Remark. In the problems to follow assume that Michelangelo **does not know** the number of erroneous transactions (all the data he has are the two Merkle trees).

Problem 11. (a) Suppose that Raphael sent Michelangelo a Merkle tree corresponding to the block on Figure 2 with exactly one of the transactions modified. Thus, Michelangelo has two different Merkle trees: the correct one and the 'falsified' one, obtained from the transactions proposed by Raphael.

(1) [5 **pts**] What is the minimal number of nodes that Michelangelo needs to compare the values at on two trees, in order to discover that the block proposed by Raphael is wrong?

(2) [5 **pts**] What is the minimal number of nodes that Michelangelo needs to compare, in order to identify the erroneous transaction?

(b) Now Raphael sent Michelangelo a Merkle tree corresponding to a block with exactly two of the transactions modified.

(1) [5 **pts**] What is the minimal number of nodes that Michelangelo needs to compare the values at on two trees, in order to discover that the block proposed by Raphael is wrong?

(2) What is the minimal number of nodes that Michelangelo needs to compare, in order to identify the erroneous transactions? We will consider different cases.

$d(A, B) = 2$ [5 **pts**]

$d(A, B) = 4$ [5 **pts**]

$d(A, B) = 6$ [5 **pts**]

⋆ **Problem 12.** [20 **pts**] Generalize part $(b)(2)$ of problem 11 to the case of binary Merkle tree with $N = 2^n$ leaves and solve it.

References

[Kob94] N Koblitz, *A course in number theory and cryptography*, 2nd ed., Graduate Texts in Mathematics, vol. 114, Springer-Verlag, New York, 1994.