Zero-Knowledge Proofs of Online Fairness

6.5610 Final Project 2026

Sejal Rathi
MIT EECS Dept.
sejalr@mit.edu

Mairin O'Shaughnessy
MIT EECS Dept.
mairin_o@mit.edu

Lillian Wang
MIT EECS Dept.
lw0328@mit.edu

Katherine Yan
MIT EECS Dept.
katyan@mit.edu

***Abstract*—Machine learning models make high-stakes decisions and learn sensitive attributes of individuals. To satisfy both model providers and users, cryptographic methods can certify fairness while maintaining privacy. We extend previous work on online zero-knowledge proofs to also prove individual fairness metrics such as ε-individual fairness and counterfactual fairness, which allows certification of a more complete notion of model fairness when coupled with existing group fairness proofs. We also show that multiclass classifiers can be efficiently proven fair in our framework, and implement a multiclass demographic parity check.**

I. Introduction

Machine learning models are being deployed in increasingly sensitive and high-stakes applications, influencing decisions in sectors from hiring decisions [1], healthcare [2], and criminal justice [3]. However, ML models, particularly those trained on biased or unrepresentative data, have also long been known [4]–[6] to be susceptible to bias and discrimination on the basis of race, sex, disability, or other such protected characteristics. As predictions issued by these models can have life-changing consequences, it becomes imperative to be able to publicly and effectively verify their fairness.

However, the desire for confidentiality serves as a major barrier to such verification. Trained models are often considered proprietary by service providers [7] and exposed only in a black-box fashion. Furthermore, the datasets used to train these models are frequently themselves confidential, as they contain sensitive or legally protected information. This difficulty has motivated a line of research into cryptographic zero-knowledge proofs of fairness, including recent works [8]–[10] each centered on different model types and fairness definitions.

In this work, we extend the online ZK proofs of fairness from the OATH framework since this work achieves a strong combination of flexibility, reliability, and scalability that we build on. We implement support for fairness definitions beyond simple demographic parity in the online setting, including for definitions of individual fairness and for multiclass outcomes. We prove the zero-knowledge properties of our implementation and conclude by evaluating our modifications to verify that scalability and reliability characteristics are preserved.

II. Related Works

FairProof [9] presents algorithms for generating fairness certificates and proving certificate correctness in zero-knowledge. They leverage geometric interpretations of fully-connected neural networks with ReLu activation as polytopes to prove an individual fairness bound. While the scheme achieves significant results for this application, white-box access to model weights is required. Further, the reduction of a neural network to polytopes requires a piecewise linear activation function [12], limiting applications. However, the distance metrics used to compute fairness bounds in ZK have potential to be adapted to various other models and situations, which we will do in this paper.

Some of the limitations of FairProof are addressed by OATH [10], which does not require access to model weights, instead relying on the queries and results of a binary classifier during online deployment to prove fairness bounds. The results presented are extensible to any arbitrary binary classifier. However, this work proves only demographic parity (DP), a group fairness metric. While models satisfying group fairness guarantee that individuals receiving a classification have the same distribution of sensitive attribute(s) as the population, Dwork et al. [18] shows that this is not a complete assessment of fairness. The authors consider cases where statistical parity is satisfied but classifications are unfair to individuals. For example, demographic parity does not exclude models which select less fit members of a sensitive group to falsely justify future discrimination. Consider the case where a classifier handles loan approvals. They could satisfy DP by approving equally many applicants from each demographic group while systematically approving the least creditworthy members of one group and the most creditworthy members of another. These groups receive the same approval rate, but individuals are not treated comparably based on their actual qualifications. Other limitations of demographic parity, such as reduced utility, where membership in a sensitive group is ignored at the expense of the success of some goal, and subset targeting, where a subset of a sensitive group is targeted, are discussed thoroughly in [18]. Enabling checks for both DP and individual fairness allows a holistic evaluation of fairness that avoids the pitfalls of either individual approach.

A. OATH: Online Fairness

OATH [10] certifies demographic parity fairness in deployed models in zero-knowledge, i.e. without leaking knowledge of client queries or model weights. Fairness audits for online data

reduce susceptibility to model switching between auditing and deployments and distribution shifts that impact fairness.

To optimize client and server interaction, the proof is split in a two-phase implementation as follows:

1) Service Phase: During the service phase, clients share commitments of queries and sensitive attributes, and the prover shares commitments of corresponding model outputs. Both additionally share signatures of these values, ensuring that if an error occurs during the service phase, the verifier can determine which signature reflects the inconsistency. Assigning blame upon failure during the service phase requires some degree of information leakage as the client must reveal the corresponding query. This phase is relatively fast as both parties only share commitments of their queries rather than complete zero knowledge proofs which are much slower.

2) Audit Phase: The audit phase involves interaction between the verifier and prover. The audit phase consists of three main stages. Stage 1 initializes IT-MAC encodings described in Section III, Stage 2 validates fairness with a ZK circuit, and stage 3 contains a collection of validity checks, also in ZK. These validity checks include: validating sensitive attribute values, ensuring the consistency of the query information given by the client and passed on by the server, and ensuring the correctness of the model output. The correctness check is handled by Mystique [11], a technique for creating ZK certificates of various ML models.

The complete algorithm is provided from Franzese et al. in Fig. 1 [10]:

Algorithm 2: OATH Zero-Knowledge Fairness Audit.

Input: *public*: the number of client queries n, fairness gap threshold θ, soundness parameter ν; $\mathcal{P}$: model M, online data $Q = \{(q_i, \alpha_s^i, o_i, r_i)\}_{i=1}^n$; $\mathcal{V}$: commitments $\{C_i\}_{i=1}^n$, sensitive attribute check strings $\{(\alpha_0^i, \alpha_1^i)\}_{i=1}^n$

Output: $\mathcal{V}$ obtains $b_{\text{pass}} \in \{0,1\}$ indicating whether M satisfies demographic parity with respect to Q.

// Step 1: Initialization

for $i \in [1, n]$ **do**

  $\mathcal{P}$ authenticates $([\![q_i]\!], [\![\alpha_s^i]\!], [\![r_i]\!], [\![o_i]\!])$;

$\mathcal{P}$ authenticates $[\![M]\!]$;

$\mathcal{P}$ authenticates $[\![c_0]\!]$ and $[\![c_1]\!]$ initialized to zero; ▷ Count positive outcomes in each demographic group

$\mathcal{P}$ authenticates $[\![n_0]\!]$ and $[\![n_1]\!]$ initialized to zero; ▷ Count individuals in each demographic group

// Step 2: Measuring Group Fairness

for $i \in [1, n]$ **do**

  $[\![s_i]\!] \leftarrow [\![q_i.\text{demographic\_attribute}]\!]$;

  $[\![b_0]\!] \leftarrow ([\![s_i]\!] == 0)$, $[\![b_1]\!] \leftarrow ([\![s_i]\!] == 1)$; ▷ Indicator bit for demographic attribute

  $[\![n_0]\!] \leftarrow [\![n_0]\!] + [\![b_0]\!]$, $[\![n_1]\!] \leftarrow [\![n_1]\!] + [\![b_1]\!]$; ▷ Update group counts

  $[\![c_0]\!] \leftarrow [\![c_0]\!] + ([\![b_0]\!] \cdot [\![o_i]\!])$, $[\![c_1]\!] \leftarrow [\![c_1]\!] + ([\![b_1]\!] \cdot [\![o_i]\!])$; ▷ Update positive outcome counts

Fairness gap computation and comparison to the threshold $[\![b_{\text{pass}}]\!] \leftarrow \left(\theta \geq \left|\frac{[\![c_0]\!]}{[\![n_0]\!]} - \frac{[\![c_1]\!]}{[\![n_1]\!]}\right|\right)$;

// Step 3: Validity Checks

$S \leftarrow$ Group-Balanced Uniform Sampling(Q, ν); ▷ An array indicating selected samples (Algorithm 6)

for $i \in [1, n]$ **do**

  // Sensitive Attribute Check

  $\mathcal{V}$ sends α_0^i, α_1^i to $\mathcal{P}$;

  $\mathcal{P}$ proves $((\![\alpha_s^i]\!] == \alpha_0^i) \oplus ([\![\alpha_s^i]\!] == \alpha_1^i)) == 1$;

  $[\![s_i']\!] \leftarrow ([\![\alpha_s^i]\!] == \alpha_1^i)$;

  $\mathcal{P}$ proves $[\![s_i]\!] == [\![s_i']\!]$;

  if $\textit{Reveal}([\![S[i]]\!]) == 1$ **then**

    // Commitment Consistency Check

    $\mathcal{P}$ proves $H(q_i||\alpha_s^i||r_i||o_i) == C_i$;

    // Inference Correctness Check

    $\mathcal{P}$ proves $[\![o_i]\!] == [\![M]\!]([\![q_i]\!], [\![r_i]\!])$ using $\mathcal{F}_{\text{inf}}$;

If any of the proofs fail, abort. Otherwise, Reveal$([\![b_{\text{pass}}]\!])$

Fig. 1: OATH audit phase pseudocode

III. Preliminaries

We first set up necessary definitions for the rest of our paper. Let $\lambda \in \mathbb{N}$ denote the security parameter. All algorithms are probabilistic polynomial-time (P.P.T.) unless stated otherwise. We write $x \xleftarrow{R} S$ to denote sampling x uniformly at random from a set S.

Definition 1 (Negligible Function [13]): A function $\mu : \mathbb{N} \to \mathbb{R}$ is *negligible*, written $\mu(\lambda)$, if for every constant $c \in \mathbb{N}$, there exists $\lambda_0 \in \mathbb{N}$ such that for all $\lambda > \lambda_0$,

$$\mu(\lambda) < \lambda^{-c}.$$

A. Interactive Proof Systems

Interactive proof systems were introduced in the mid 1980s by Goldwasser, Micali, and Rackoff [14], who proposed a view of verification as an interaction between a prover algorithm P and a verifier algorithm V.

Both the verifier and prover receive the input of the problem instance. They exchange messages sequentially, each computing the next message as a function of the transcript so far. At the end of the interaction, the verifier decides whether to accept or reject the proof.

Definition 2 (Interactive Proof System [15]): An *interactive proof system* for a language L consists of an interactive P.P.T. verifier algorithm V and an interactive (possibly inefficient) prover algorithm P, which exchange a series of messages $m_1, \ldots, m_k$, with each message computed as a function of all previous messages. Notably, the verifier's computations may also depend upon private random bits not revealed to the prover. We write $(P, V(r))(x) = 1$ to denote the event that the verifier V, with private randomness r, accepts after interacting with P on joint input x. The following two properties are required:

1) **Completeness.** For every $x \in L$:

$$\Pr_r\big[(P,\, V(r))(x) = 1\big] \;\geq\; \tfrac{2}{3}.$$

If the statement is true and the prover is honest, the verifier accepts with significant probability.

2) **Soundness.** For every $x \notin L$ and every (malicious and possibly all-powerful) prover P^*:

$$\Pr_r\big[(P^*,\, V(r))(x) = 1\big] \;\leq\; \tfrac{1}{3}.$$

If the statement is false, a cheating prover can only convince the verifier to accept with very small probability.

Remark 3: The constants $\frac{2}{3}$ and $\frac{1}{3}$ are arbitrary. By repeating the protocol λ times and accepting if and only if at least $\frac{\lambda}{2}$ rounds accept, the Chernoff bound gives completeness $1-\mu(\lambda)$ and soundness $\mu(\lambda)$.

B. Zero-Knowledge Proofs

While completeness and soundness ensure that an interactive proof system is correct, they do not address what information is revealed during the interaction. Ideally, the verifier should learn nothing beyond the validity of the statement being proven. This notion is captured by zero knowledge, which formalizes the idea that whatever the verifier sees during the protocol could have been generated without interacting with the prover at all.

Definition 4 (Zero Knowledge [15]): An interactive proof system (P, V) for a language L is *zero knowledge* if for all

(malicious and possibly all-powerful) verifiers V^*, there exists a probabilistic polynomial-time (P.P.T.) algorithm Sim (the "simulator") such that for all $x \in L$,

$$\{\mathrm{View}_{V^*}(P(x), V^*)\} \approx \{\mathrm{Sim}(x)\},$$

where $\mathrm{View}_{V^*}(P(x), V^*)$ denotes the view of the verifier V^* in its interaction with the prover P on input x, namely its internal randomness together with all messages it receives during the protocol.

C. Information-Theoretic Message Authentication Codes (IT-MACs)

In OATH [10], secure computation relies on *information-theoretic message authentication codes* (IT-MACs) to authenticate values in the field $\mathbb{F}_p$ using the extension field $\mathbb{F}_{p^r}$. These take the form:

$$M_x = K_x + \Delta \cdot x \in \mathbb{F}_{p^r}$$

One party (in our case, the prover) holds the value $x \in \mathbb{F}_p$ and its corresponding MAC tag $M_x \in \mathbb{F}_{p^r}$, where $\mathbb{F}_{p^r}$ is the polynomial extension field for $\mathbb{F}_p$, i.e. $\mathbb{F}_{p^r} \cong \mathbb{F}_p[X]/f(X)$ where $f(X)$ is an irreducible polynomial of degree r.

The other party (in our case, the verifier) holds the global key $\Delta \xleftarrow{R} \mathbb{F}_{p^r}$ and the message key $K_x \xleftarrow{R} \mathbb{F}_{p^r}$.

An IT-MAC-authenticated value is denoted with double brackets as $[\![x]\!]$, and computation on these values is denoted as e.g. $[\![z]\!] \leftarrow [\![x]\!] + [\![y]\!]$. IT-MACs, in addition to satisfying correctness and unforgeability properties of traditional MACs, also satisfy:

1) **Hiding**, as Δ and K_x are sampled uniformly, and so a malicious verifier V gains no knowledge of x.
2) **Binding**, as a malicious prover P attempting to forge a message-tag pair for some $x' \neq x$ must compute $M_{x'} = K_x + \Delta \cdot x' = M_x + \Delta(x' - x)$. As P does not know Δ, it can only attempt to successfully guess it with probability $1/p^r \in \mathsf{negl}(r)$.

Thus, IT-MACs may be used as a secure commitment scheme.

IT-MACs are linearly homomorphic, meaning addition over committed values can be computed without interaction between prover and verifier. Wolverine builds on this foundation [16] with an efficient interactive protocol for generating and computing on these authenticated values. In particular, Wolverine allows for zero-knowledge evaluation of arbitrary boolean or arithmetic circuits, with addition gates computed locally and multiplication gates require only one round of communication between prover and verifier, followed by a batch verification at the end of the protocol. We use Wolverine as the underlying ZKP system for all constructions that follow.

IV. Problem Formulation

A. Parties & Inputs

Our construction primarily focuses on modifications to the OATH scheme's audit phase [10], showing that different measures of online model fairness can be computed in ZK. As such, we will assume that query, answer, and sensitive attribute commitments and validity checks, as well as information transfer between parties, are handled correctly and reliably.

We define three parties: a model provider P of classifier M, a set of clients $\{C_i\}_{i=1}^n$, and a verifier V. Clients send queries q_i of a defined length m to provider P, who returns a classification $o_i = M(q_i)$. In the general set-up, P claims a value of a fairness metric, and the verifier validates this value over a set of online client queries in zero-knowledge.

B. Threat Model

We assume that the weights of M are proprietary and only known by P. This motivates a cryptographic approach to fairness auditing to protect against a model-switching attack by a malicious P as described in [10]. When model weights are private, a malicious provider may demonstrate fairness on an offline "fair" model, but deploy a model that does not meet a fairness criterion. This threat is addressed by auditing fairness against a deployed model using real client queries, making it impossible to use a different model for auditing and deployment.

A second phenomenon motivates the need for the three party set-up. It would be more straightforward to have an independent verifier which evaluates the model on a static dataset, but machine learning models are vulnerable to distribution shift [17]. A distribution shift occurs when a model that was fair and accurate on initial deployment becomes unfair on current queries. Periodically dynamically evaluating fairness using current client queries ensures that fairness is correctly validated and may help in identifying distribution shifts that affect fairness.

V. Methods

With this setting established, we extend the OATH audit to capture two complementary definitions of individual fairness into the OATH setting: ε-individual fairness and counterfactual fairness. Both definitions share a common structure: rather than imposing aggregate constraints over a dataset, they impose relational constraints over pairs of inputs. Specifically, we evaluate model behavior under structured perturbations and enforce consistency of predictions across such pairs.

A. Local ε-Individual Fairness

We first consider local ε-individual fairness, which defines fairness through a neighborhood condition. This metric says that two clients whose non-sensitive features are "sufficiently close" should receive identical predictions, regardless of their sensitive attributes. By using a distance threshold ε, this definition allows for a flexible degree of similarity based on a defined distance metric.

Definition 5 (Local ε-Individual Fairness - ℓ_2 Norm): Let $S_i \in \{0, 1\}$ be the sensitive attribute of client C_i. Let each client's query to the predictor $\hat{Y}$ be equal to $q_i = S_i || x_i$, where q_i is a vector of length m. $\hat{Y}$ is locally individually fair with respect to the ℓ_2 distance between queries if for all pairs q_i, q_j:

$$||x_i - x_j||_2 \leq \varepsilon \implies \hat{Y}(q_i) = \hat{Y}(q_j)$$

We choose the ℓ_2 norm as our example distance metric since our implementation represents queries as bit vectors. This allows us to compute distances using bitwise rather than arithmetic operations, which is especially significant in the squaring operations. However, other choices of distance may be chosen based on the application. Any distance metric $d : X \times X \to \mathbb{R}$, must satisfy $d(x, x') \geq 0$, $d(x, x') = d(x', x)$, and $d(x, x) = 0$ [18].

We now describe how local ε-individual fairness is verified within the ZK audit. At a high level, the verifier must establish two things for each pair (q_i, q_j): first, whether the non-sensitive features x_i and x_j fall within ℓ_2 distance ε of each other; and second, if they do, whether the model returned identical predictions on both queries. Both checks are performed against IT-MAC committed values; the verifier never sees the queries or outputs in the clear, only their commitments. The ZK proof therefore needs to demonstrate, for every qualifying pair, that the committed outputs are equal, and for every non-qualifying pair, that the distance condition was correctly evaluated and the pair was legitimately excluded from the equality check.

Upon completion, the verifier obtains a single bit b_{pass} indicating whether local ε-individual fairness held across all n client queries in the audit window. Importantly, a result of $b_{\text{pass}} = 0$ does not identify which pairs violated the condition. The proof reveals only that at least one violation exists, preserving individual query confidentiality.

Algorithm 1 Zero-Knowledge Proof of Local ε-Individual Fairness

Input: public: number of clients n, fairness threshold ε, query length m; P: online data $Q = \{(q_i, s_i, o_i,)\}_{i=1}^{n}$
Output: V obtains $b_{\text{pass}} \in \{0, 1\}$ indicating if ϵ-individual fairness is satisfied over Q
for all pairs $([[q_i]], [[q_j]]) \in Q$ **do**
  if $||[[x_i]] - [[x_j]]||_2 \leq \varepsilon$ **then**
    $[[b_{ij}]] \leftarrow [[o_i]] == [[o_j]]$
  else
    $[[b_{ij}]] \leftarrow 1$
  end if
end for
Compute $[[b_{pass}]] \leftarrow \mathbf{AND}_{i,j}[[b_{ij}]]$
If validity checks [10] pass, `Reveal`$([[b_{pass}]])$

B. Counterfactual Fairness

Counterfactual fairness represents a special case of ε-individual fairness where $\varepsilon = 0$. It requires that a model's prediction remain strictly unchanged when a sensitive attribute is modified while all other features are held exactly fixed, i.e. . This ensures that the prediction is not directly dependent on protected characteristics such as race or gender.

Definition 6 (Counterfactual Fairness [19]): Let $q_i = S_i||x_i$ be our model inputs, where S_i is the sensitive bit and x_i represents the non-sensitive features. Let $M(q_i)$ denote a predictor. The predictor M is *counterfactually fair* if for all x_i,

$$M(0||x_i) = M(1||x_i).$$

Equivalently, for any pair of inputs q, q' that differ only in the sensitive attribute,

$$M(q) = M(q').$$

This definition enforces that predictions are invariant to changes in the sensitive attribute when all other information is unchanged. In practice, this is evaluated by constructing paired inputs that differ only in A, running inference on both, and checking whether the outputs are equal.

Whereas local ε-individual fairness compares arbitrary pairs of client queries, counterfactual fairness requires a more structured pairing: each real query is matched with a synthetic counterpart in which only the sensitive attribute has been flipped. The verifier therefore does not construct pairs from the observed query set alone, it must also obtain or verify the counterfactual queries and answers $Q' = \{q'_i, o'_i\}_{i=1}^{n}$, where q'_i is identical to q_i in all non-sensitive features x_i but differs in the sensitive attribute S_i. Since the counterfactual queries are synthetic constructions, the proof must additionally certify that each q_i and q'_i are well formed and correctly related. This introduces an additional verification burden relative to the local fairness case.

We enforce this within the OATH framework by issuing paired inference queries to the deployed model: one using the original client input q_i and one using its counterfactual q'_i. The provider P runs the model on both and commits to both outputs. The verifier then checks equality of the committed outputs in zero knowledge, learning only whether the fairness condition holds and not the underlying feature values of either query.

Upon completion, the verifier again obtains a single bit b_{pass}. As with local ε-individual fairness, a result of $b_{\text{pass}} = 0$ reveals only that some counterfactual pair produced differing predictions, without identifying which client was affected. Unlike the ε-individual fairness case, however, every client contributes exactly one pair to the audit, there is no distance threshold to tune, and the number of proof obligations scales linearly in n rather than quadratically. The fairness check circuit is therefore cheaper, but this gain is offset by the need to commit and validate an entirely separate set of n counterfactual queries Q', each of which will require its own IT-MAC authentication and validity checks under the OATH protocol, which is overhead that the ε individual fairness case avoids entirely by operating only on actual client query commitments.

C. Multiclass Fairness Verification

We also seek to expand the applicability OATH's group fairness check by exploring its multiclass variant and examining how performance and security results scale with multiple classes. Specifically, we assume the model output can take on values $\{0, 1, ..., k-1\}$. There are two primary sites that are

Algorithm 2 Zero-Knowledge Proof of Counterfactual Fairness

Input: public: number of clients n, fairness threshold ε; P: online data $Q = \{(q_i, s_i, o_i,)\}_{i=1}^{n}$, counterfactual online data $Q' = \{(q'_i, s'_i, o'_i,)\}_{i=1}^{n}$
Output: V obtains $b_{\text{pass}} \in \{0, 1\}$ indicating if counterfactual fairness is satisfied over Q
for all $i \in [1, n]$ **do**
  $[\![b_i]\!] = [\![o_i]\!] == [\![o'_i]\!]$
end for
Compute $[\![b_{pass}]\!] \leftarrow \textbf{AND}_i[\![b_i]\!]$
If validity checks [10] pass, `Reveal`($[\![b_{pass}]\!]$)

affected by such a change: the inference correctness check and, depending on the fairness definition, the fairness check. This paper focuses primarily on the fairness check stage, but we also briefly discuss the correctness check.

1) Correctness Check: The correctness check involves evaluating an inference verification circuit. This process is abstracted and described in [11] which provides circuits proving machine learning functions in zero knowledge. This includes matrix multiplication as well as both sigmoid and softmax functions. Empirical results show that verifying sigmoid is roughly two orders of magnitude faster than softmax over 10 classes (2.1 ms compared to 209 ms). However, matrix multiplication with 1024×1024 matrices requires roughly 2 seconds to verify, which is likely the greater bottleneck to the inference check circuit. Therefore, as long as the models remain structurally similar besides the final layer, scaling the number of classes should not greatly change the runtime of the inference verification circuit.

2) Fairness Check: For both individual fairness metric definitions, the query outputs are only taken into account when checked for equivalence with "close" queries. In these cases, increasing the number of discrete classes requires no algorithmic change.

Meanwhile, demographic parity would need to be evaluated for each class and combined in some way. We use the definition in [21].

Definition 7 (ε-Demographic Parity fairness): A classifier M is ε-fair with respect to sensitive attribute s over a set of queries Q if and only if:

$$max_i \, |Pr[M(q) = i | s = 0] - Pr[M(q) = i | s = 1]| \leq \varepsilon$$

where $q \xleftarrow{R} Q$.

This definition of demographic parity means the fairness verification scales linearly with the number of classes. Also note that we use the maximum value over all the classes due to simplicity of the implementation. Alternative definitions include ensuring the sum of the values is less than ε, or ensuring that some large percentage of classes have demographic parity less than ε. Studying effects of different definitions of multiclass demographic parity is left to future work. An implementation of this modification to fairness verification is given in algorithm 3.

Algorithm 3 Zero-Knowledge Proof of k-class ε-Demographic Parity Fairness

Input: public: number of clients n, fairness threshold ε, number of classes k; P: online data $Q = \{(q_i, a_s^i, o_i, r_i)\}_{i=1}^{n}$, commitments $\{C_i\}_{i=1}^{n}$
Output: V obtains $b_{\text{pass}} \in \{0, 1\}$ indicating if ϵ-multiclass demographic parity fairness is satisfied over Q
P authenticates $[\![b_{pass}]\!]$ initialized to 1.
for all $i \in [0, k-1]$ **do**
  $[\![c_0]\!], [\![c_1]\!]$ initialized to zero.
  $[\![n_0]\!], [\![n_1]\!]$ initialized to zero.
  for all $j \in [1, n]$ **do**
    $[\![s_j]\!] \leftarrow [\![q_j.\text{demographic\_attribute}]\!]$;
    $[\![b_0]\!] \leftarrow ([\![s_j]\!] == 0)$; $[\![b_1]\!] \leftarrow ([\![s_j]\!] == 1)$;
    $[\![n_0]\!] \leftarrow [\![n_0]\!] + [\![b_0]\!]$, $[\![n_1]\!] \leftarrow [\![n_1]\!] + [\![b_1]\!]$;
    $[\![o]\!] \leftarrow ([\![o_j]\!] == i)$
    $[\![c_0]\!] \leftarrow [\![c_0]\!] + ([\![b_0]\!] \cdot [\![o]\!])$, $[\![c_1]\!] \leftarrow [\![c_1]\!] + ([\![b_1]\!] \cdot [\![o]\!])$
  end for
  $[\![b_{pass}]\!] \leftarrow [\![b_{pass}]\!]$ **AND**
    $(\theta \geq |[\![c_0]\!]/[\![n_0]\!] - [\![c_1]\!]/[\![n_1]\!]|)$
end for
If validity checks [10] pass, `Reveal`($[\![b_{pass}]\!]$)

The main change is in the addition of an outer for-loop dedicated to each output class and the calculation of b_{pass}. The additional complexity in the fairness check circuit affects the soundness. Specifically, the bound on probability of false positives in the verification process is primarily affected by the number of multiplication gates in the circuit. Since AND operations require multiplication gates and the demographic parity calculation is repeated k times, the number of multiplication gates as well as the runtime should scale linearly with the number of classes. Assuming the number of gates is small compared to the size of the prime field being operated over, the soundness of the fairness check should remain negligible. The relationship between soundness and the number of multiplication gates is discussed further in Section VI.

VI. Evaluation & Discussion

A. Soundness Argument

The algorithms described for each fairness metric form a Wolverine [16] ZK circuit where P holds IT-MAC-authenticated values for the input wires (e.g. queries, outcomes, counterfactual outcomes), and P and V evaluate the circuit over these input values to produce output bit b_{pass}. V accepts the proof iff $b_{pass} = 1$.

For such a ZK circuit, [16] bounds the probability that V wrongly accepts a proof where P does not hold a valid witness to be at most

$$\binom{CB + c}{B}^{-1} + \frac{1}{p^r} + \varepsilon_{open} \leq \mathsf{negl}(C, B, r)$$

where C is the number of multiplication gates in the circuit, B, c are tunable protocol parameters, p^r is the size of the

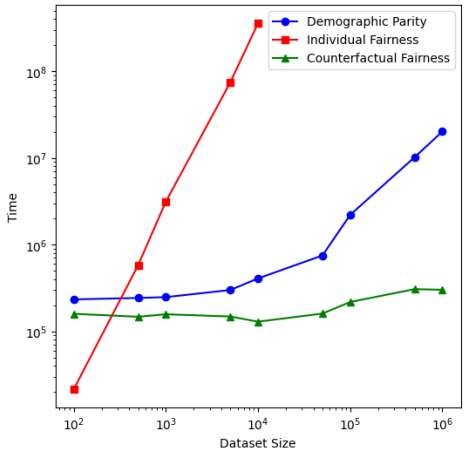


Fig. 2: Dataset size plotted against ZK proof runtime in microseconds for demographic parity (DP), individual fairness (IF), and counterfactual fairness (CF) metrics. Note that a log-log axis is used. IF proofs are more computational expensive to compute for large datasets than DP and CF.

extension field $\mathbb{F}_{p^r}$, and ε_{open} is the negligible soundness error of one of the underlying sub-protocols.

The correctness of the input values themselves are verified by other sections of the OATH framework as shown in Step 3 in Figure 1.

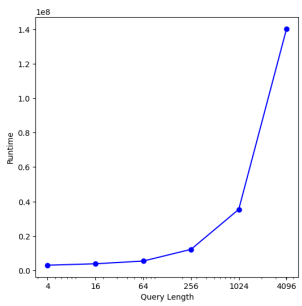
B. Runtime Performance

We evaluate the runtime performance of the zero-knowledge proof protocol for each fairness metric. Implementations were written using EMP-toolkit [20] and experiments run by locally simulating both parties on a Windows machine with an Intel Core i7 processor. To isolate the cost of the fairness check protocol only, a set of idealized query sets suitable for each algorithm were generated. Runtime scaling with database size, query length, and number of classes can be seen in figures 2, 3a, 3b. As expected from algorithmic design, IF proofs are more computationally expensive to compute than proofs for DP or CF.

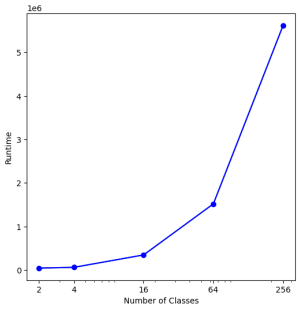
Runtime for IF and CF do not scale with number of classes, and runtime for DP and CF do not scale with query length.

C. Catch Probability by Audit Size

It is worth noting that the most expensive audit stages of the OATH framework are the correctness and consistency checks that validate that outcomes are reported correctly, i.e. that $M(q_i) = o_i$ for all (q_i, o_i) query-answer tuples in our query dataset Q. OATH [10] introduces an optimization where, while the fairness metric is computed over the full dataset Q, correctness is only checked for a randomly-sampled subset $Q' \subseteq Q$ of all query-answer tuples, thereby reducing total proof runtime at the cost of certainty in cheating detection, as a prover may misreport outcomes in $Q \setminus Q'$ and escape detection with some probability. OATH [10] bounds this probability as a function of $v = |Q'|$ and $\epsilon = |X_h - X_m|$, where X_h is the



(a) Individual Fairness proof runtime in microseconds for varying query lengths.



(b) Multiclass DP proof runtime in microseconds for varying number of classes.

Fig. 3: Runtime scales linearly with query length for IF and linearly with number of classes for DP. Note that a log-y axis is used for both graphs.

fairness gap calculated under honest outcomes, and X_m is the fairness gap reported by a cheating P.

As this bound depends on the fairness metric being audited, any discussion of alternative fairness metrics should discuss corresponding implications for this bound.

1) Demographic Parity: We restate the definitions and bound provided by [10], as it provides the structure with which we will make similar arguments for other fairness metrics. [10] defines X_h as the demographic parity gap on an honestly-reported query dataset Q_h, and X_m as the demographic parity gap measured by V on an arbitrary query dataset Q_m provided by a malicious P.

$$X_h = \left|\frac{\sum_{(q_i,o_i)\in Q_h} o_i \cdot I_0(s_i)}{\sum_{(q_i,o_i)\in Q_h} I_0(s_i)} - \frac{\sum_{(q_i,o_i)\in Q_h} o_i \cdot I_1(s_i)}{\sum_{(q_i,o_i)\in Q_h} I_1(s_i)}\right|$$

$$X_m = \left|\frac{\sum_{(q_i,o_i)\in Q_m} o_i \cdot I_0(s_i)}{\sum_{(q_i,o_i)\in Q_m} I_0(s_i)} - \frac{\sum_{(q_i,o_i)\in Q_m} o_i \cdot I_1(s_i)}{\sum_{(q_i,o_i)\in Q_m} I_1(s_i)}\right|$$

Where $I_0(s_i) = 1$ if the $s_i = 0$ and $I_1(s_i) = 1$ iff $s_i = 1$.

With $\epsilon = |X_h - X_m|$, [10] derives the probability of catching a cheating prover is at least $1-(1-\frac{\epsilon}{2})^v$, where v is the number of queries sampled.

This bound extends to our multiclass definition of DP, as the possible difference in fairness gap caused by misreporting a single outcome in the multiclass model is bounded by the possible difference in a binary model.

2) Counterfactual Fairness: We demonstrate a similar property for counterfactual fairness by creating metrics X_h and X_m that represent the proportion of queries in an honestly-provided and maliciously-provided dataset that demonstrate counterfactual unfairness, i.e.

$$X_h = \frac{\sum_{(q_i,o_i,o_i')\in Q_h} \mathbb{1}(o_i \neq o_i')}{|Q_h|}$$

$$X_m = \frac{\sum_{(q_i,o_i,o_i')\in Q_m} \mathbb{1}(o_i \neq o_i')}{|Q_m|}$$

Since ϵ is then directly the proportion of queries that have been modified, we arrive at $p_{\text{catch}} = 1 - (1-\epsilon)^v$.

3) Individual Fairness: We define X_h and X_m to be the proportion of query pairs that demonstrate unfairness, i.e.

$$X_h = \frac{\sum_{((q_i,o_i),(q_j,o_j))\in Q_h^2} \mathbb{1}(||x_i - x_j||_2 \leq \varepsilon \ \wedge \ o_i \neq o_j)}{|Q_h|^2}$$

$$X_m = \frac{\sum_{((q_i,o_i),(q_j,o_j))\in Q_m^2} \mathbb{1}(||x_i - x_j||_2 \leq \varepsilon \ \wedge \ o_i \neq o_j)}{|Q_m|^2}$$

To change $\epsilon|Q|^2$ query pairs, a malicious prover must have changed at least $\sqrt{\epsilon}|Q|$ queries, resulting in $p_{\text{catch}} \geq 1 - (1 - \sqrt{\epsilon})^v$.

D. Metric limitations

The three fairness definitions described in this work, while all suitable for use with the OATH framework, present different advantages and disadvantages based on end user requirements. Some necessary considerations are enumerated below.

1) Demographic Parity: Demographic Parity provides a middle ground in fairness-check runtime. However, as addressed in Section II, group fairness metrics come with their own particular limitations. As is established in fairness literature [18], [23], a classifier may satisfy population-level metrics while still remaining unfair on an individual level, and in fact attempts to optimize purely for between-group fairness may exacerbate within-group disparities. These limitations motivate the use of individual-level fairness metrics, which impose constraints at the level of individual predictions rather than aggregate statistics.

2) Local ε-Individual Fairness: As implemented in this paper, the individual fairness audit has the slowest overall fairness-check runtime, scaling quadratically with the size of the query dataset. IF, however, does offer a greater flexibility, as the ε-threshold may be tuned to suit particular requirements. Furthermore, while our implementation uses the ℓ_2-norm as a distance function between queries, IF may be implemented for any reasonable distance function – provided, of course, that one exists, as many applications may not come with a natural conception of distance. For applications where such a distance function is available and a tunable similarity threshold is desirable, IF offers a compelling alternative to group-level metrics. When a stricter, more structured notion of individual fairness is required, counterfactual fairness may be appropriate.

3) Counterfactual Fairness: This presents the fastest fairness proof generation, as its audit circuit uses only boolean operations and scales linearly with the size of the query dataset, D. However, although the fairness audit itself is cheap, the need to process both factual and counterfactual queries q_i, q_i' means that other parts of the OATH protocol such as the correctness and consistency checks operate over a dataset size of $2D$. As these checks dominate end-to-end proof runtime [10], we cannot take counterfactual fairness as unambiguously cheaper. Furthermore, while the concept of a counterfactual query is straightforward when the sensitive attribute is binary (e.g. veteran status), it becomes more awkward to define for other characteristics. Even for binary attributes, some applications may find counterfactual fairness to be insufficient as a fairness guarantee, and may turn to a more general IF for stronger guarantees.

VII. Conclusion

In this work, we show that efficient online zero-knowledge proofs of fairness of proprietary machine learning models are extensible to individual fairness metrics and multi-class models. This broadens the applicability of the OATH framework in high-stakes settings such as hiring, healthcare, and criminal justice, where individual-level fairness guarantees and protection of sensitive client data are critical, and where model providers require assurance that model parameters and structure remain hidden throughout the audit process. A limitation of our work is that bounds for individual fairness must be tuned or adjusted based on specifics of the model, the data inputs, and the applications. Future work to determine the value of ε-individual fairness that a model satisfies over a set of inputs would be beneficial in detecting the severity of distribution shift. Further, we would like to explore how more complex inputs such as images could be proven to be individually fair since the sensitive attributes would be less clearly defined.

Contributions & Acknowledgment

Sejal worked on implementation of counterfactual fairness as well as exploration of other fairness metrics, specifically Equalized Odds and kNN Local Fairness. Mairin implemented local individual fairness and made figures. Katherine conducted the experiments to collect runtime data and proved catch bounds. Lillian implemented local individual fairness and demographic parity for multiclass models and analyzed soundness bounds with additional classes. All four team members contributed to the writing and editing of this paper.

We would also like to thank course instructor Srini Devadas, as well as TAs Simon Langowski, Kevin He, Xiaochen Zhu, and Jophy Ye for their excellent teaching and guidance throughout this course and project.

References

[1] S. Wall, "LinkedIn's job-matching AI was biased. The company's solution? More AI.," MIT Technology Review, Jun. 23, 2021. https://www.technologyreview.com/2021/06/23/1026825/linkedin-ai-bias-ziprecruiter-monster-artificial-intelligence/.

[2] H. Javed, Hafiz Abdul Muqeet, T. Javed, Atiq Ur Rehman, and R. Sadiq, "Ethical Frameworks for Machine Learning in Sensitive Healthcare Applications," IEEE Access, vol. 12, pp. 1–1, Jan. 2023, doi: https://doi.org/10.1109/access.2023.3340884.

[3] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine Bias," ProPublica, May 23, 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[4] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," ACM Computing Surveys, vol. 54, no. 6, pp. 1–35, Jul. 2021, doi: https://doi.org/10.1145/3457607.

[5] A. Lambrecht and C. E. Tucker, "Apparent Algorithmic Bias and Algorithmic Learning," SSRN Electronic Journal, 2020, doi: https://doi.org/10.2139/ssrn.3570076.

[6] K. Waddell, "How Algorithms Can Bring Down Minorities' Credit Scores," The Atlantic, Dec. 02, 2016. https://www.theatlantic.com/technology/archive/2016/12/how-algorithms-can-bring-down-minorities-credit-scores/509333/.

[7] I. Lederer, R. Mayer, and A. Rauber, "Identifying Appropriate Intellectual Property Protection Mechanisms for Machine Learning Models: A Systematization of Watermarking, Fingerprinting, Model Access, and Attacks," IEEE transactions on neural networks and learning systems, pp. 1–19, Jan. 2023, doi: https://doi.org/10.1109/tnnls.2023.3270135.

[8] A. S. Shamsabadi et al., "Confidential-PROFITT: Confidential PROof of FaIr Training of Trees," OpenReview, 2023. https://openreview.net/forum?id=iIfDQVyuFD.

[9] C. Yadav, A. R. Chowdhury, D. Boneh, and K. Chaudhuri, "FairProof : Confidential and Certifiable Fairness for Neural Networks," arXiv.org, 2024. https://arxiv.org/abs/2402.12572.

[10] O. Franzese, A. S. Shamsabadi, C. Luck, and H. Haddadi, "Secure and Confidential Certificates of Online Fairness," arXiv.org, 2024. https://arxiv.org/abs/2410.02777.

[11] C. Weng, K. Yang, X. Xie, J. Katz, and X. Wang, "Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning," USENIX Security 21, 2021. https://www.usenix.org/conference/usenixsecurity21/presentation/weng.

[12] S. Black et al., "Interpreting Neural Networks through the Polytope Lens," arXiv.org, 2022. https://arxiv.org/abs/2211.12312

[13] S. Devadas and 6.5610 Staff, "One-Way Hash Functions," MIT 6.5610 Lecture Notes, Lecture 1, Massachusetts Institute of Technology, Cambridge, MA, USA, Feb. 2, 2026.

[14] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems (extended abstract)," in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, R. Sedgewick, Ed., Providence, Rhode Island, USA, May 6–8, 1985, pp. 291–304, ACM, 1985.

[15] S. Devadas and 6.5610 Staff, "Interactive Proofs and Zero Knowledge," MIT 6.5610 Lecture Notes, Lecture 11, Massachusetts Institute of Technology, Cambridge, MA, USA, Mar. 11, 2026.

[16] C. Weng, K. Yang, J. Katz, and X. Wang, "Wolverine: Fast, Scalable, and Communication-Efficient Zero-Knowledge Proofs for Boolean and Arithmetic Circuits," Cryptology ePrint Archive, 2020. https://eprint.iacr.org/2020/925.

[17] S. M. Kulinski and D. I. Inouye, "Towards Explaining Distribution Shifts," Jul. 2023.

[18] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. "Fairness through Awareness". Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS 12).

[19] M. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 4069–4079.

[20] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. "EMP-toolkit: Efficient MultiParty computation toolkit", github, 2016. https://github.com/emp-toolkit.

[21] C. Denis, R. Elie, M. Hebiri, and F. Hu, "Fairness guarantee in multi-class classification", arXiv.org, 2023. https://arxiv.org/abs/2109.13642

[22] I. Damgard, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology – CRYPTO 2012*, Lecture Notes in Computer Science, vol. 7417, pp. 643–662, Springer, 2012.

[23] S. Caton and C. Haas, "Fairness in Machine Learning: A Survey," ACM Computing Surveys, vol. 56, no. 7, Aug. 2023, doi: https://doi.org/10.1145/3616865.