An Analysis of Enrollment and Query Attacks on Hierarchical Bloom Filter-based Biometric Systems

Sumaiya Shomaji, *Student Member, IEEE,* Pallabi Ghosh, *Student Member, IEEE,* Fatemeh Ganji, *Member, IEEE,* Damon Woodard, *Senior Member, IEEE,* and Domenic Forte, *Senior Member, IEEE*

***Abstract*—A Hierarchical Bloom Filter (HBF) -based biometric framework was recently proposed to provide compact storage, noise tolerance, and fast query processing for resource-constrained environments, e.g., Internet of things (IoT). While security and privacy were also touted as features of the HBF, it was not thoroughly evaluated. Compared to the classical BFs, the HBF uses a threshold parameter to make robust authentication decisions when the HBF encounters noise in the biometric input which one would think might lead to security issues. In this paper, the attack vectors that could compromise the HBF security by increasing the false positive authentication of non-members and by leaking soft information about enrolled members are explored. With quantitative analyses, HBF-based biometric system security under these well-defined attack vectors is evaluated and it is concluded that the framework is more difficult to attack than the classical Bloom Filter. Further, experimental results show that soft biometric information is also kept private.**

***Index Terms*—Bloom Filter, Biometric Authentication, Security and Privacy, Access Control.**

I. Introduction

THE Internet of Things (IoT) contains billions of connected devices that could provide access control services in smart buildings, smart homes, etc. In such applications, biometrics are an attractive technology due to their convenient ability to leverage an individual's unique physiological or behavioral properties. However, developing the ideal framework for a biometric system is challenging, especially if there are numerous users involved. For example, consider a large-scale smart building security system that relies on facial recognition. A large central database would be required to store genuine facial templates of the buildings employees and visitors. Each IoT camera would need to capture and then transmit the faces of individuals in its view as queries to this database. Such an approach suffers from three primary drawbacks. First, it would take a huge amount of space to store all the employee images. Second, the search time to match each query from thousands of face data would encounter significant latency. Finally, template theft from the central database could be leveraged to perform various biometric template attacks such as access control circumvention [15] or fraud.

The popular choices for biometric indexing protocols include data structures (e.g., k-d tree, Locality sensitive hashing or LSH, etc.) that support fast comparison between query individual's template and enrolled templates. These are storage-efficient and suitable for fast search/indexing. However, several recent instances have demonstrated that no matter whether the methods consider raw templates or encoded templates on the system, they cannot be completely relied on in terms of assuring privacy of the biometric templates and resistance against authentication performance manipulation. For instance, the encoding-based privacy-preserving templates can still suffer from data recovery attacks [20]. The encryption-based feature protection templates require additional storage and computational resources. Therefore, for resource-constrained devices, they might not be useful at all. Similarly, due to the advancement in computer vision and reverse engineering, extraction of sensitive and confidential information from classifiers or data structures like k-NN, k-d trees or CNN model are also increasing alarmingly. One additional major drawback of these systems is for high dimensional data, their classification/authentication performance is not satisfactory (k-d tree), some require cloud data for storing the templates and/or processing the authentication task (privacy-preserving CNN). In brief, there is a lack of biometric access-control solution that provides storage efficiency, fast query handling, and template security/privacy.

Bloom Filter (BF) -based structures are very promising candidates with the potential to meet the three aforementioned requirements. BFs are probabilistic data structures that store data in a compressed manner and enable very fast membership query processing. If a BF employs a secure, one-way hashing technique on the biometric templates of genuine users during the enrollment and membership query handling, it becomes computationally infeasible for an attacker to breach/reverse engineer its data or falsely authenticate. However, BFs have one significant challenge when it comes to biometrics. Specifically, the hash-based comparisons between enrolled templates and query template require an exact one-to-one match. Due to the inherently fuzzy nature of biometric templates, even a single bit flip will cause a drastic change in the hash output [9]. In our prior work [31], we proposed a biometric indexing system referred to as the Hierarchical Bloom Filter (HBF) framework. In addition to the storage efficiency and fast query processing, analytical formulas were derived that allowed the HBF to tolerate the inherent fuzziness/noise of biometric templates/queries.

While HBF offers several desirable features for a biometric authentication system, the security and privacy of this framework are yet to be analyzed. However, there has been some

S. Shomaji, P. Ghosh, D. Woodard, and D. Forte are with the Department of Electrical and Computer Engineering, University of Florida (email: shomaji@ufl.edu, pallabighosh@ufl.edu, dwoodard@ece.ufl.edu, and dforte@ece.ufl.edu)

F. Ganji is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (email:fganji@wpi.edu)

Manuscript received June 13, 2021; revised August XX, 2021.

prior work to analyze the BF security, e.g., [11]. Comparing the HBF to a classical BF, a noteworthy question is whether or not its noise tolerance introduces new security/privacy issues. In this paper, we define appropriate threat models for HBF-based biometric systems and perform a comprehensive security analysis against those threats. The major contributions are:

- We define the adversarial enrollment attack which involves malicious increase of false positive (FP) authentication.
- We define two types of adversarial query attacks where the first type involves malicious increase of false positive (FP) authentication and the second type involves extraction of soft biometric information.
- For adversarial enrollment attack and FP increase-based query attack, we derive mathematical expressions for representing the probability of launching both the attacks. Compared to expressions for the classical BFs, we find that the HBF-based system is more difficult to attack.
- Since the mathematical formulation is non-trivial for information-leakage-based query attacks, we conduct several experiments for a face indexing system. We considered gender, age, and ethnicity as soft information for leakage experiments. In all cases, we find that the attacker's success is no better than a random guess.
- We also consider the appropriate privacy metrics established by the scientific community to measure the level of protection offered by the HBF.

The rest of the paper is organized as follows. In Section II, the background behind the HBF-based biometric system's framework is discussed. In Section III, the privacy metrics considered for determining the protection provided by the HBF are introduced. Section IV and V discuss the adversarial enrollment and adversarial query models (respectively) which are applicable for the HBF-based biometric system. The results of several experiments evaluating the security of HBF for multiple adversarial goals are demonstrated in Section VI. In Section VII, the existing work related to the biometric indexing, privacy and security are discussed along with their salient differences with the HBF-based system. Finally, Section VIII concludes the paper.

II. Hierarchical Bloom Filter (HBF) Background

Since an HBF is a collection of BFs organized into multiple levels, it is necessary to understand the basics of BFs before describing the HBF itself. Thus, Sections II-A and II-B discuss the background of BFs and HBFs, respectively.

A. BF-Based Data Structures for Fast Query

A BF is a space-efficient, probabilistic data structure which can be used to quickly determine whether a data input is not in a set or possibly in it [4, 32]. The space efficiency is achieved at the price of introducing the false positives (FP). However, the FPs can be kept very low by configuring the design parameters of the BF as per the developer's requirements. There are two major stages in BF: enrollment and query. The four major parameters associated with the construction of a BF during the enrollment stage are: number of samples/users for enrollment (n), false-positive probability or priori ($FP_{BF,pri}$),

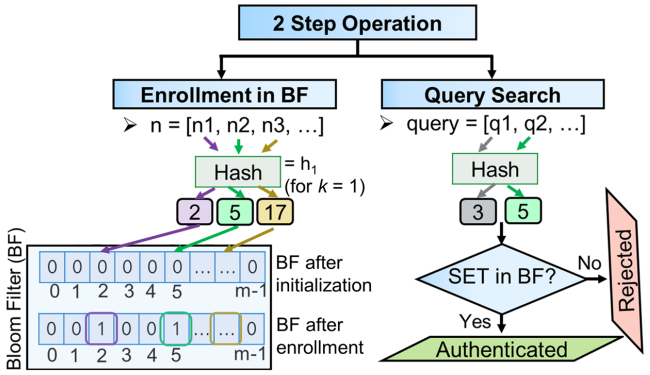

Fig. 1: Steps taken to (a) enroll a new element in the BF, and (b) process the queries for membership search.

number of hash functions (k), and size of the BF in units of bits (m). A list of all the parameters associated with BF and HBF is shown in Table I. Note that, there are two additional false-positive related concepts considered in this paper, they are: posteriori and no. of false positives. To illustrate, the priori or false positive probability ($FP_{BF,pri}$) is set by the developer during the initialization of the BF. However, by adopting different adversarial models (introduced in Section IV and V), an attacker can cause the deviation of the $FP_{BF,pri}$ set by the developer. Thus, the altered false positive probability (FP_{adv}^{BF}) is determined after the query processing (for a particular adversarial model denoted as 'adv') and referred as posteriori. Finally, to introduce the no. of false positives, there are instances of false positive authentication after the query processing and the total number of such instances is the number of FPs. The two step operation for a legacy BF is introduced below.

Enrollment of samples in BF. As shown in Figure 1, the first step is the enrollment of the samples (e.g., $n1$, $n2$, and $n3$) in the BF. Before enrollment, optimum values for all the BF parameters should be determined. For a given n and $0 \leq FP_{BF,pri} \leq 1$, m and k can be determined using the analytical formulas in [32]. A zero vector with a length of m bits is then initialized as the BF. Next, the enrollment process for a sample begins by performing k different hashes. For enrolling the sample into the BF, the indices equal to the output of each hash are SET as "1" and are referred to as SET bits. For example, in Figure 1, $k = 1$ is assumed and therefore all the samples are hashed once by hash function h_1. For enrolling $n1$, its hash output $h_1(n1)$ is found to be "2". Therefore, the second index in the BF is SET, thus enrolling n_1. This process is repeated for all the samples and finally, the n number of samples are enrolled. Note that the numerical value of the h_1's may exceed m. Therefore, after generating the hash outputs (h_1s in this example), a modulus function is applied to them to ensure that they fit within the range of $[0, m-1]$.

Membership query processing in BF. To perform query processing, the BF receives the query of a sample and hashes it k times with the same hash functions as in the enrollment stage. Once the hash outputs are generated, the indices of the BF equal to the generated hash output values are accessed and checked to see whether they are all SET bits or not. If so, then the query sample is considered as a member of the BF,

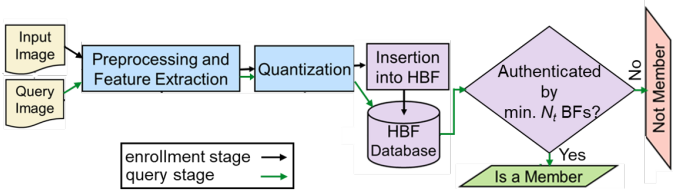

Fig. 2: Flow of HBF operation during enrollment and query.

whereas if at least one of the bits in those indices is not found SET, the query sample is "rejected" as a non-member of the BF. This explains why query sample q_1 will be rejected but q_2 authenticated in Figure 1.

Since hashing is an integral part of BF structures, it cannot be applied to noisy data such as biometric templates. The diffusion property of the hash algorithm will reject any query that is not identical to one of the enrolled templates. For instance, suppose a user is enrolled in BF with template t_1. During query processing, she provides another template, t_1' which differs from t_1 due to the presence of measurement noise and other variations (e.g., aging). If the legacy BF is considered for the membership check of t_1', the user will never be authenticated because the hash output of t_1 and t_1' will be different. Therefore, we proposed the HBF-based framework for a biometric system that will retain all the appealing features of a classical BF and at the same time offer noise tolerance [31].

B. HBF-Based Data Structures for Fast, Noise-tolerant Query

An HBF is a data structure consisting of multiple BFs organized in a hierarchy. Such a structure has been used as a solution for sub-string matching problems. To illustrate for the case of biometrics, the HBF checks if a part of a template, is enrolled in a BF or not. This sort of checkup is useful for biometric systems because biometric templates are fuzzy. Thus, instead of looking for a perfect match, if the majority of parts of a query template match with that of an enrollment template, the HBF will decide that a query person is an enrolled person (i.e., member) or otherwise, an unenrolled person (i.e., non-member). Like legacy BF, the HBF also operates in two steps which are discussed below.

Enrollment of samples in HBF: To begin enrollment, as shown in Figure 2, the input biometric templates from users are pre-processed and features are extracted. The pre-processing varies with the type of biometric modality [31]. For example, if the input templates are digital face images of users, pre-processing includes face detection, alignment, and masking to generate a normalized image for every user. Next, the collected features are quantized to a binary template. The binary templates are then split into non-overlapping fixed-size blocks. Note that, the splitting and insertion steps are not shown in Figure 2 to keep it simple and hence demonstrated in a separate figure.

The templates are now ready for enrollment in the HBF. Assume that, there are N number of BFs with IDs as BF_{ID} $(ID = 0, 1, \ldots, N-1)$ distributed in d levels in the HBF and the length of each block in i-th level is l_i bits $(i = 0, 1, \ldots, d-1)$. Now once these blocks from the templates are prepared, each block from every enrollment sample is independently inserted into its associated BF (see a toy example in Figure 3). To illustrate, at the bottom level, the first l_0-bit block from all the samples (figure shows a single sample for simplicity) will be inserted in BF_0, the second l_0-bit blocks will be enrolled in BF_1, and so on. This is how the BFs at leaf level are filled. Since HBFs conduct matching of sub-strings or blocks, this may result in an increase in FP due to combinations of blocks that are incorrectly reported as being in the BF [32]. To tackle this problem, along with the blocks, their concatenations are also inserted in the BFs at levels upper than leaf level (d=0). This is mainly the reason why there are multiple levels (d) and different block lengths (l_i) for each level at HBF are considered. Therefore, resuming from d=1, w number of subsequent blocks from the previous level are concatenated for sample enrollment in the current level. Thus, the more upward we move in the HBF, the larger the l_i becomes. At the root level, all of the blocks from the sample get concatenated by this time, meaning the entire template is considered as a block, and thus the BF at the root level is the same as the legacy BF.

Similar to BF, it is also a requisite to determine all the HBF parameters to initialize the HBF structure to prepare it for enrollment. For n members and false-positive probability or priori for HBF ($FP_{HBF,pri}$), the size of the BFs (m), optimal number of hash functions (k), d, l_i, and number of subsequent blocks concatenated in each layer of the HBF (w) are determined. For brevity, the formulas can be found in [31].

Membership query processing in HBF: During query processing, the query sample goes through the same preprocessing, feature extraction, and quantization step as the enrolled

TABLE I: Notations and their descriptions.

Notation	Description
d	# layers in the HBF
k	# hash functions in a BF
l_i	# bits in a block of a template in i-th layer of the HBF
m	The length of BF in bits
n	# enrollment template
N	total # BFs in an HBF
w	# subsequent blocks concatenated in each layer of the HBF
W	# SET bits in a BF
$m - W$	# unSET bits in a BF
FP	false positive
$FP_{BF,pri}$	priori (false positive probability during initialization) in BF
FP_e^{BF}	posteriori (false positive probability after query) in BF during enrollment attack
FP_q^{BF}	posteriori (false positive probability after query) in BF during query attack
$FP_{HBF,pri}$	priori (false positive probability during initialization) in HBF
FP_e^{HBF}	posteriori (false positive probability after query) in HBF during enrollment attack
FP_q^{HBF}	posteriori (false positive probability after query) in HBF during query attack
N_{auth}	# authenticating BFs in HBF
BF_i	i-th BF in HBF

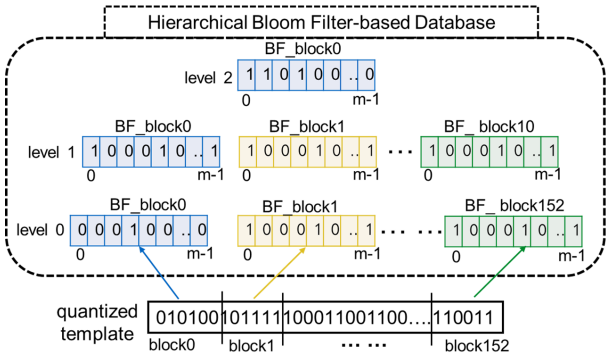

Fig. 3: Schematic of an HBF. The templates are split into blocks. The smallest blocks are inserted in the BFs located at the lowest level, i.e., level 0. The whole string is inserted in the BF at the highest level d (here, $d = 2$).

templates (see Figure 2). Once the binary template of the query sample is obtained, it is also split into blocks (same as HBF enrollment step). Now, considering each block as a sub-string of query sample, a membership query processing is performed in the associated BF. Assuming the number of BFs authenticating the query sample in the HBF is defined as N_{auth}, if N_{auth} exceeds a threshold (N_t), then the query sample is considered as a member. Else, it is a non-member.

N_t and $FP_{HBF,pri}$ should be chosen very carefully since they are directly involved with the decision-making process. However, in contrast to common scenarios, where BFs or HBFs are employed to store the data, in biometrics-related applications, we have to deal with noisy templates during both enrollment and membership query processing events. Hence, one should consider the following probabilities in HBF-based biometric systems: true positive (TP_{TH}), true negative (TN_{TH}), false positive (FP_{TH}), and false negative (FN_{TH}) [31]. The expressions for determining all these parameters are thoroughly discussed in [31]. Further, since the objective of this paper is to analyze the security feature of HBF, we avoid the discussions on authentication performance and refer the reader to [31].

The vulnerabilities and threats applicable for probabilistic data structures including BFs have been comprehensively studied in the literature [8, 11]. A salient difference between the BF and HBF is the HBF's tolerance to noise/fuzziness of input queries, which might introduce a security/privacy concern. In the next three sections, we aim to study this. We will start by introducing the privacy metrics from the literature that are applicable in this paper and then move on to the adversarial models in the context of face recognition[1] for the BF and HBF. Then the security features of HBF against those models are discussed.

III. Privacy Assessment Metrics

The 'amount' of protection provided by our HBF-based system should be measured by privacy metrics as proposed in relevant literature (for a comprehensive survey, see [33]).

[1]This is merely for ease of discussion and relation to the experiments section. We expect our conclusions in this section to generalize to other biometric modalities as well.

There are several metrics proposed in [33] from which we select metrics based on the following criteria:
1. The metrics which are applicable in database domain are considered in this study since HBF is a framework for representing biometric database.
2. The metrics which have a similar adversarial model or goal are considered.
3. If the selected metrics based on above two criteria involve input parameters which can be easily determined for HBF as well, we directly adopt that for measuring HBF's privacy. If not, then we adapt the privacy metric with respect to the input parameters available for HBF.

According to these criteria, the following metrics are considered in our analyses.

A. Adversary's Success Probability:

This metric, in general, indicates how likely it is for the adversary to succeed. For the HBF, the success of the attacker lies in causing the posteriori or FP positive authentication probability exceed the priori set by the developer during the enrollment and query attack. The lower the success probability is, the higher privacy can be achieved. The success probability indicates the privacy of the individuals on average. This metric relies on the definition of an adversary model considered for an attack.

B. Information Surprisal

In order to measure how much information is leaked when the attacker makes a query with a specific feature. The metric has been already defined in the social network area [7]; however, it has found application in other privacy-related studies [33]. More precisely, it quantifies how much information is contained in a specific outcome x of a random variable X. In this regard, X represents the user's templates with a specific feature, whereas $p(x)$ is the probability of that specific feature within the set of all enrolled users.

C. Error

The error-based metrics measure the correctness of adversary's estimate. High correctness and small errors correlate with low privacy. The "percentage incorrectly classifier" metric measures the percentage of incorrectly classified users within the set of all users.

The metrics highlighted in this section are used to assess the adversary's impact on users' privacy of a system coming under attacks introduced in Sections IV and V. More precise definitions of these metrics and their corresponding adversary models are also given in those sections.

IV. Adversarial Enrollment Attack

The attack models with the ability to violate the integrity and confidentiality of HBF functionality can be broadly classified into two categories: (i) adversarial enrollment attack and (ii) adversarial query attack [11]. In this section we discuss the adversarial enrollment attack.

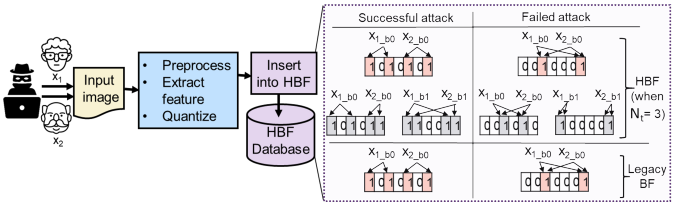

Fig. 4: Enrollment attack where attacker tries to maximize number of SET bits during enrollment by inserting samples that select distinct bits. Attack fails when distinct bits are not chosen. For successful attack, in legacy BF only k bits need to be SET whereas k SET bits are needed in $\geqslant N_t$ BFs for the HBF.

A. Adversarial Model

The adversarial enrollment attack takes place during the enrollment stage and it is assumed that the items (e.g., fingerprints, irises, faces, etc.) to be enrolled in the HBF are chosen by the attacker. The attacker can only insert biometric templates (e.g., face biometrics considered in this study). If the attacker inserts a general item, e.g., text string, the enrollment will immediately fail because the pre-processing step right after the input acquisition step (see Figure 2) will detect the absence of the particular biometric template and halt. The goal of this attack is to deliberately increase the FP rate to one higher than expected by the designer's analytical expressions. Such a higher FP rate is desirable to increase the probability of getting a non-member authenticated by the HBF. In the worst-case scenario, the adversary can completely fill the HBF causing a DoS attack, where any arbitrary query is authenticated as a member [11].

The adversarial enrollment attack can be performed by pollution attack. To perform the pollution attack, first, an empty HBF is considered. Next, the adversary starts inserting n items in such a way so that the maximum number of bits can be SET for those n items. During the attack, the attacker tries to ensure the maximization of SET bits. The more there are SET bits in the BFs of the HBF, the more chances of the HBF authenticating unenrolled samples (non-members). Examples are shown in Figure 4 where the attacker succeeds and fails to enroll input face images x_1 and x_2 in such a way they result in distinct (unSET) bits to be SET and cause SET bit maximization in the BFs. Note that the attacker has only one attack point – the input image receiver module.

B. Metric

An attack is considered successful if the SET bits are maximized. For an HBF with N BFs and an authentication threshold of N_t BFs, a successful enrollment attack is performed if the attacker meets the SET maximization criterion for at least N_t BFs. Since the attack directly affects the probability of a false positive, it is measured to demonstrate the attacker's success. To better align with the definition of the privacy metric "Adversary's Success Rate" (SR) provided in [33], we employ the $priv_{SR,e}$ defined below.

$$priv_{SR,e} = \Pr(\text{FP after the query during enrollment attack}) = FP_e^{HBF} \geq \tau_{SR,e} \quad (1)$$

Here the parameter $\tau_{SR,e}$ is a pre-defined one set by the system designer that is the priori for HBF.

C. Security Analysis

In two ways the security of the HBF against the enrollment attack is assessed, namely: conceptually and mathematically. The conceptual assessment provides our intuition about the practical impediments for the attacker and the mathematical assessment derives an expression for the probability of a successful attack.

Conceptual Assessment: As shown in Figure 4, the framework receives a natural input image as an enrollment template (e.g., from a camera) and it is ready for insertion into the BF/HBF only after it has gone through preprocessing, feature extraction, and quantization. Therefore, the only attack point is the sensor/camera which the attacker can hijack to submit her crafted/desired images. During an enrollment attack, the goal is to maximize the SET bits. This occurs if the input images result in distinct bits (instead of common bits) for each enrolled user. However, the choice of bit allocation depends on the quantized template, and the quantized template depends on preprocessing and feature extraction of the natural image. Therefore, it is almost impossible for an attacker to ensure this except by brute forcing on a separate testbed beforehand – this would not be a scalable attack.

Mathematical Assessment: We start with the attack against the legacy BF. Assume there are n users for enrollment and k hash operations done to enroll each of them into an m-bit sized BF. The goal of the attacker is to ensure the maximization of the SET bits during the enrollment process. The maximization of SET bit is achieved by obtaining k distinct bits from the unSET (bits which are '0' in BF) bits. For example, when the first user is enrolled, the BF is empty (all the bits are unSET), thus the attacker can choose any k distinct bits to get SET. Assuming that the total number of SET bits is $W = k$ after enrollment of one user, if the attacker now wants to enroll the second user in the BF, she has to choose k distinct bits for the second user from the $m - W$ remaining unSET bits. Thus, the probability of enrollment attack for a single legacy BF is defined by the probability of causing bit maximization through the enrollment process. In other words, by adopting the adversarial enrollment model, the attacker is causing a deviation in the $FP_{BF,pri}$ set by developer. Thus, the altered false positive probability (FP_{adv}^{BF}) aka posteriori is measured and referred to as the probability of enrollment attack (FP_e^{BF}). It can be represented by [11]:

$$FP_e^{BF} = \frac{{}^{m-W}C_k}{m^k} \quad (2)$$

where ${}^nC_k = \frac{n!}{k!(n-k)!}$. Here, $0 \leq W \leq nk$ where $W = 0$ means no one has been enrolled yet. When n users are enrolled and k distinct bits are SET for each of them, W is maximized with a value of nk.

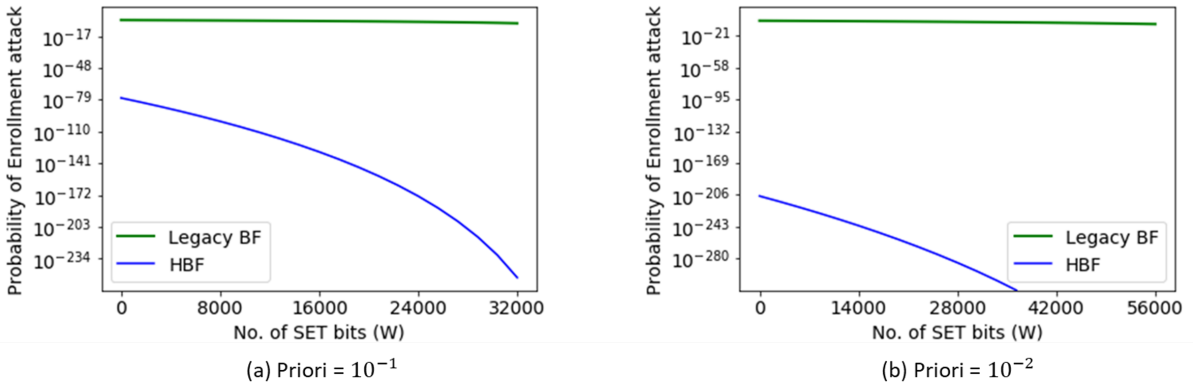

Fig. 5: Comparing HBF's resistance against enrollment attack with that of legacy BF in terms of posteriori (considering equation 2 and 3) at two different parameter settings based on priori.

Now, consider an HBF with N BFs and an authentication threshold of N_t BFs. In order to conduct a successful enrollment attack in HBF, the attacker has to meet the SET maximization criterion for at least N_t BFs individually. There are multiple combinations of individuals BFs where the attack can succeed. Therefore, the probability of enrollment attack for an HBF is the posteriori of the HBF (FP_e^{HBF}) after the enrollment attack, which can be represented by:

$$FP_e^{HBF} = \frac{\sum_{i=N_t}^{N} ({}^{N}C_i)({}^{m-W}C_k)^i}{(m^k)^N} \tag{3}$$

In this equation, the term ${}^{m-W}C_k$ denotes how many ways an attacker can choose k bits from $m-W$ unSET bits, where the order of selection does not matter. In order to launch a successful attack, the attacker has to meet this criterion for at least threshold (N_t) BFs. Among the N BFs, the N_t BFs can be chosen in ${}^{N}C_i$ ways where i is in $[N_t, N]$ range. Therefore, the numerator considers a summation over the range $[N_t, N]$ to represent required bit selections for the enrollment attack. To compute the probability of a successful attack, this term should be multiplied by $1/(m^k)^N$ denoting the probability of setting a bit in an HBF (see Equation (2) for a single BF).

To illustrate the difference between FP_e^{HBF} and FP_e^{BF}, we use Equations (2) and (3) to verify numerically how resistant against enrollment attack is the HBF. Assuming $n = 8,000$ users and $N = 56$ BFs, we vary W and measure FP_e^{BF} and FP_e^{HBF} for two different false positive probabilities or priori: $\tau_{SR,e} = 10^{-1}$ and $\tau_{SR,e} = 10^{-2}$. As shown in Figure 5, while increasing W, both the FP_e^{BF} and FP_e^{HBF} decreases. An increase in W (SET bits) means a decrease in $m-W$ (unSET bits). Thus, when the number of unSET bits decreases, the chances of finding indices where unSET bits are present also decrease, and consequently, the enrollment attack probability is reduced. For legacy BF, it is only a single BF where the attacker has to find the unSET bits for enrolling an item. However, for the HBF, this has to be done for at least N_t BFs, which is more difficult. Thus, the probability of enrollment attack in HBF is much less than enrollment attack in BF (i.e., $FP_e^{HBF} \ll FP_e^{BF}$). Additionally, based on this discussion, for the given $\tau_{SR,e}$ and the configuration of the system (i.e., n and N), as $priv_{SR,e} < \tau_{SR,e}$, the adversary is unsuccessful as well.

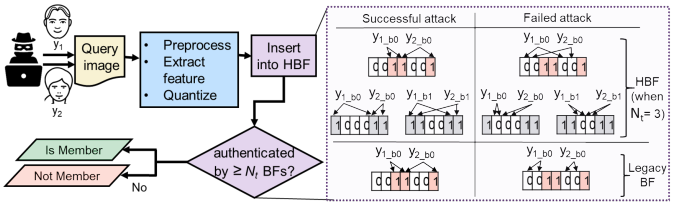

Fig. 6: Adversarial query attack for FP authentication where attacker attempts false authentication by providing query templates (y_1 and y_2) whose hash outputs overlap entirely with any of the SET bits from enrollment. For successful attack, in legacy BF the overlap is required only for the single legacy BF itself, whereas for HBF, the overlap must be occurring in $\geqslant N_t$ BFs.

V. Adversarial Query Attack

In this section, we discuss the adversarial query attack and the security measures taken by the HBF against it.

A. Adversarial Model

The adversarial query attack is performed during the query stage. For this attack, three assumptions are made. First, the HBF is always initialized and maintained by a trusted party (e.g., HBF developer). Therefore, the attacker cannot directly alter the bits in the HBF that got SET or not-SET during the enrollment stage. Second, the identity and confidential soft information (e.g., gender, age, ethnicity) about the enrolled individuals are not public; hence unknown to the attacker. Third, the authentication decisions by the HBF and the individual BFs are made public and thus known to the adversary. Note that, similar to adversarial enrollment attack, the attack point for adversarial query attack is also a single one. Here, it is the query image receiver module.

The two potential query attack goals that we consider are:

- **FP Authentication Goal:** Here, the goal of the attacker is to achieve a false positive authentication of unenrolled (non-member) query samples. To initiate this attack, the attacker crafts diverse images as query templates in such a

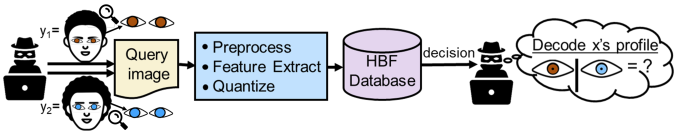

Fig. 7: Adversarial query attack for information leakage (attacker trying to leak information e.g., age, gender, ethnicity etc. about enrolled user by observing the authentication decision made by HBF and individual BFs about her provided query templates y_1 and y_2).

way so that after going through all the template processing (as shown in Figure 2) steps, the output of each of the hash function is overlapped with a SET bit in the HBF. In doing so, the query template will get authenticated even if the query individual was not enrolled. Using this information, an attacker could execute a presentation attack against the biometric system by creating a spoof (fake finger, mask, etc.) based on the falsely authenticated template. Examples are shown in Figure 6 where the attacker succeeds and fails to craft query images y_1 and y_2 in such a way that while getting compared, they only choose those bits in the BFs that are already SET by other templates.

- **Information Leakage Goal:** Here the goal of the attacker is to infer soft biometric information about the enrolled individuals (e.g., skin, eye or hair color, height and weight, gender, race, wrinkles, etc.) by performing queries[2]. As shown in Figure 7, during this attack, the attacker submits samples from multiple eye color classes (e.g., y_1 contains brown eyes and y_2 has blue eyes) to the HBF and observes the authentication decision made by the HBF. If one of the classes have samples that are authenticated more than that of other, then the attacker can guess that someone with a specific eye color is more likely to be contained in the HBF. By attempting queries with different classes-based on gender, height, hair color, etc., the attacker may be able to determine if an individual is in the HBF.

B. Metrics

Adversary's Success Rate (SR): According to the adversary model mentioned above, the following metrics indicating the success of this attack can be defined.

$$priv_{SR,q} = \Pr(\text{FP after the query during query attack}) = FP_q^{HBF} \geq \tau_{SR,q} \quad (4)$$

with $\tau_{SR,q}$ being a system designer's pre-defined parameter, which is the posteriori (false positive probability after query) for the HBF. Similar to Adversary's SR discussed in the literature [33], this metric is useful to assess the adversary's success of getting query templates authenticated even if the query individual has not been enrolled.

Information Surprisal (IS): After launching a query attack, the adversary could be capable of extracting soft information about the enrolled individuals. To observe how successful this attacker could be, the metric IS is defined below. In the context of this attack, defined in this paper, X represents a well-crafted query made by the attacker (e.g., querying a template with a specific gender), whereas $p(x)$ is the probability of a particular feature within the set of all enrolled users. $priv_{IS}$ indicates the amount of privacy loss after the attack.

$$priv_{IS} = -\log_2 p(x) \quad (5)$$

Percentage Incorrectly Classified (PIC): Another metric applicable for reporting the success of an attack, and consequently, the loss of privacy, is the percentage of incorrectly classified individuals [33]. For our HBF-based system, the attacker can be interested in revealing the characteristics of individuals. For instance, guessing the age, gender, or ethnicity of a target in HBF can be aimed by the adversary. The percentage of incorrect guesses U' within the set of all users U (i.e., the total number of enrollments) is used to define the following metric.

$$priv_{PIC} = U/U' \quad (6)$$

C. Security Analysis of HBF

The adversarial query attack is of two types – FP authentication-based and information leakage-based. The security analysis of the HBF framework against the former type is conducted by both conceptual analysis and mathematical analysis below. However, for the information leakage type attack, the mathematical analysis is nontrivial. Hence, we instead conduct multiple experiments in Section VI to observe the HBF's resistance against information leakage.

Conceptual Assessment: During the FP authentication-based query only attack (Figure 6), the attacker's goal is to cause the overlapping of query image's bit allocation with the SET bits of enrolled items. For the same reason mentioned in the conceptual analysis of HBF against enrollment attack, ensuring the overlapping of query image's bit allocation with the SET bits of enrolled items in HBF cannot be guaranteed by providing tailored query images during the query-only attack. For the information leakage attack as well, it is very difficult for the attacker to correlate the query sample with enrolled samples. In brief, there is only one attack point for the attacker which is the query image reception for both types of query attacks (see Figures 6 and 7) and the attacker can attempt to alter the bits only via this input. This is due to the fact that after the reception of the query image, it goes through multiple processing steps and finally gets hashed. Hence, the bit allocation in the HBF for even slightly different templates can be quite divergent. Therefore, whether attempting to get false positive authentication or leak information by providing tailored query items, we expect that achieving either of the goals are extremely difficult for the attacker.

Mathematical Analysis: During the FP authentication-based query-only attack, the adversary has no control over the enrollment steps. Rather, she keeps on crafting query templates (which are most likely unenrolled) and submitting them into

[2]While soft biometrics are not unique to a specific subject, they can be used to infer information about the enrolled users, including their identities or other privacy concerns. For example, if ECG is used as a biometric, inference of cardiac arrhythmias in the ECG of the subjects contained in the HBF could result in discrimination.

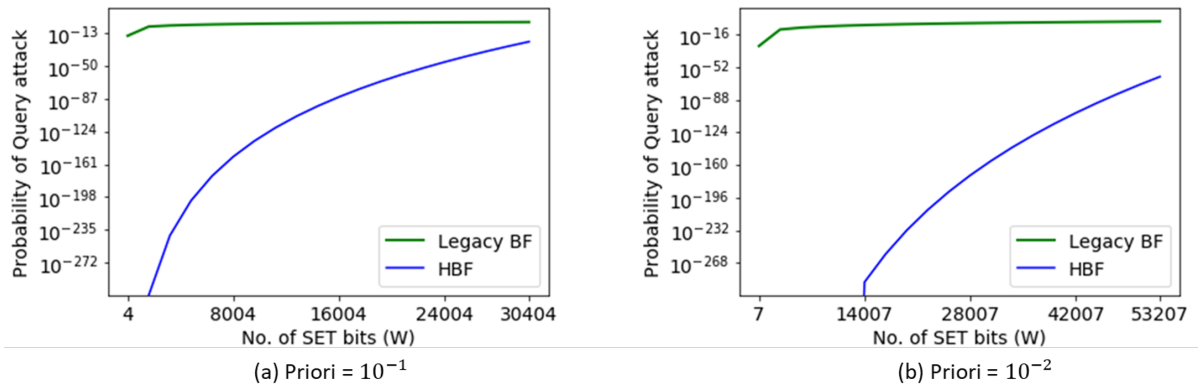

Fig. 8: Comparing HBF's resistance against query attack with that of legacy BF in terms of posteriori (considering equation 7 and 8) at two different parameter settings based on priori.

the biometric system so that at least one of her templates gets accepted. Thus, the probability of query-only attack can be defined as the probability of FP occurring for unenrolled query data. Note that, the FP authentication occurs when all of a non-member's k hash outputs overlap with any of the SET bits. In fact, the same SET bit can even be obtained k times too in order to cause FP authentication for a non-member. The probability of FP authentication-based query attack in a legacy BF for a query item (y) is defined by the probability of causing the k number of mod($h_i(y)$)(s) getting overlapped with any of the W bits during the query process. In other words, by adopting the FP-authentication based adversarial query model, the attacker is causing a deviation in the $FP_{BF,pri}$ set by the developer. Thus, the altered false positive probability (FP^{BF}_{adv}) aka posteriori is measured and referred to as the probability of FP authentication-based query attack (FP^{BF}_{q}). This is given by [11]:

$$FP^{BF}_{q} = \left(\frac{W}{m}\right)^{k} \tag{7}$$

Here the range for W is $1 \leq W \leq nk$ bits, where $W = 1$ refers to enrolling at least one item, whereas W is maximized when n users are enrolled.

Now, we consider an HBF with N BFs and an authentication threshold of N_t BFs. In order to conduct a successful query attack in HBF, there are two crucial facts. First, the attacker has to cause the overlap of the query item with any of the W bits in at least N_t of the individual BFs. Second, the N_t BFs can be chosen in different combinations. Considering these two facts, the probability of FP-authentication based query attack is the posteriori of the HBF (FP^{HBF}_{e}) after the query attack. This can be represented by:

$$FP^{HBF}_{q} = \frac{\sum_{i=N_t}^{N} ({}^{N}C_i)(W^k)^i}{(m^k)^N} \tag{8}$$

Here, the numerator considers the events, where an attacker can successfully launch the query attack, whereas the term $1/(m^k)^N$ shows the probability of each event. Note that, this criterion for the attack should be fulfilled by at least N_t BFs.

Using the expressions for FP^{HBF}_{q} and FP^{BF}_{q} for HBF and BF, we vary parameter settings to numerically verify whether HBF is more resistant against query attacks or not. The same setting as the previous section is used ($\tau_{SR,q} = 10^{-1}$ and $\tau_{SR,q} = 10^{-2}$). As shown in Figure 8, while increasing W, both the FP^{BF}_{q} and FP^{HBF}_{q} increase. When W increases, it works in support of the attacker because the goal of the attacker is to find SET bits and make her query template overlaps with the W bits during the membership-query processing stage. Therefore, the increase in W leads to an increase in query attack probability in both the legacy BF and the HBF structure. However, if considered individually, the FP^{BF}_{q} is very small and, for HBF, this probability has to be achieved by at least N_t BFs – this makes the overall probability for HBF which is FP^{HBF}_{q} much smaller than FP^{BF}_{q}. Moreover, according to the above analysis, for the given design of the system (i.e.,$\tau_{SR,e}$, n and N), as $priv_{SR,q} < \tau_{SR,q}$, the adversary is not successful (for the results related to $priv_{IS}$, see Section VI-B).

Important Takeaways: Due to the inclusion of the threshold parameter N_t in the authentication process for HBFs (i.e., its noise tolerance), one might draw the conclusion that the security and privacy of the legacy BF would be better than HBF. This is because the HBF is designed to authenticate queries that are "close to" enrolled samples. To give a better understanding, consider that if the query template is too noisy but its respective individual is an enrolled user, the proposed HBF might have to lower the N_t value. In such a scenario, if there is any query data which is not enrolled, it can still get authenticated by mistake because of the low threshold. The legacy BF does not have any such threshold and is therefore very sensitive to any noise in the input query. After our thorough analysis of the adversarial models both conceptually and mathematically, we are confident that even when the value of N_t is low, getting authenticated by N_t BFs is of very small probability. Thus, the HBF actually provides better security and privacy than the classical BF while handling noise. However, none of the frameworks are resistant to spoofing attack although adding a liveness detector step in the preprocessing module can contribute to resist the attack. This might be considered as a future research direction.

VI. Experimental Results

In this section, we conduct multiple experiments in two categories to evaluate HBF's security measures against information leakage-based adversarial query attack. We consider a face indexing/membership system. The first category shows whether the HBF as a whole, i.e., the HBF's membership decision is impacted if a query template has any similarity in its feature with that of the enrolled templates. The second category showcases whether an individual BF within the HBF shows an inclination for authentication towards samples from a specific class. The types of soft biometric classes that we consider are gender, ethnicity, and age. In Sections VI-A and VI-B, the experiments under both categories are discussed in detail. First, however, our experimental setup is discussed.

Dataset: The dataset considered in this paper is "UTKFace" [37]. There are four major reasons for the choice. First, the dataset is a challenging dataset because it has samples from 5 different ethnic groups whose age ranges from 1-116 years, and enough samples are available for both male and female classes. Second, according to [18], the dataset contains samples from three different sources: web, celebrity dataset CACD [6], and Morph [27] where the samples for the first two are obtained in the wild and the third contains samples taken in a controlled environment. Often many studies perform well for a dataset which has samples collected in a controlled environment but the same study deviates a lot performance-wise when the approach is applied to wild or uncontrolled data. Since this dataset has a blend of various data, it will be fair to use this to realize whether it can perform well in most/all of the situations or not.

Third, the samples involve large variations in expression, pose, illumination, and style; hence, adding more challenges to the dataset. Fourth, this is also a public dataset thereby allowing others to reproduce the results mentioned in this paper. Note that, originally the dataset has 23,708 samples with age, gender, and ethnicity annotations. However, since there are some samples from the wild, we follow steps from recent works [2, 34] by processing the samples before beginning the experiments. The processing includes wrong label detection, removal of wrongly labeled samples, and removal of ditto images, etc. Then, as shown in Figure 2, dedicated face processing techniques (including face detection, alignment, and face normalization) are performed. After all the processing, we move forward with a subset of 19,106 images for our experiments.

HBF implementation: The experiments begin with designing a framework of HBF for a given dataset. According to the theory of HBF, for a given n and a desired FP, the associated parameters m and k can be computed for HBF. Once these parameters are known, N and N_t are determined in a recursive manner using the formulas provided in [31]. The HBF allows now the enrollment of the users after their biometric templates are pre-processed. After the completion of the user enrollment, the membership query process begins. Since the HBF and its parameters are public, during the query processing step, the verifier can know about three vital information from the HBF: (i) whether the provided query data is authenticated or not by HBF, (ii) how many BFs authenticated the query data (N_{auth}) (iii) what are the indices of the authenticating BFs (BF_i) [31]. According to the operating principle of HBF, if $N_{auth} \geq N_t$, only then the query data is considered a member. There is a fourth parameter too which can also be revealed, i.e., the number of enrolled items. However, this does not affect the security or privacy of the enrolled users because the revelation of the number of enrolled items does not help the attacker to infer any sensitive information about the individual enrolled items. Thus, the fourth parameter is not considered as a security threat and has been deliberately ignored from any consideration in further experiments.

Note that, the focus of this paper is not to evaluate the authentication performance of HBF. This is already covered in [31] where we used FP, FN, TP, and TN to demonstrate its performance and explained how these metrics can be manipulated by tweaking N and N_t to achieve a desirable authentication performance. Thus, the choice and discussion of N_t is out of scope for this paper and thus the performance metrics-based results are deliberately ignored in this paper for the experiments. Instead, the emphasis is made on the factors which are directly related to the security and privacy of the HBF: (i) total number of BFs that authenticate query templates from diverse classes when the HBF has enrolled only the templates from a particular class and (ii) individual BF's probability towards authenticating a template from a particular class when HBF has enrolled templates from multiple classes. Note that both the factors are directly related to N_{auth} and BF_i respectively.

Interestingly, the two experiments discussed above (whether the HBF as a whole is impacted if a query template has any similarity in its feature with that of the enrolled templates and whether an individual BF within the HBF shows an inclination for authentication towards samples from a specific class) are directly helping in determining these two factors. Thus, by conducting the experiments in the two categories mentioned in Sections VI-A and VI-B, the security measures provided by HBF can be measured. Along with the FP rate, since the size of the template also does not change throughout the experiments, we obtain $N \approx 56$ for UTKface using the formulas in [31]. Note that, to guarantee the statistical soundness of the results, a 10 fold cross-validation approach has been considered for each of the experiments.

A. Exploiting N_{auth} for Information Leakage

The goal of this experiment is to verify whether the HBF is more inclined towards authenticating queries from a particular class that shares some common features with the enrolled templates than the query samples from other classes that do not. If this happens, then the attacker can try submitting samples from multiples classes to the HBF, and for whichever sample class she gets more authentication outputs, she will realize that the templates enrolled in the HBF share similar features with the authenticated query samples. Thus, she can relate those features to the enrolled users in the HBF which contradicts the HBF's privacy objective. We conducted individual experiments to confirm this where we considered

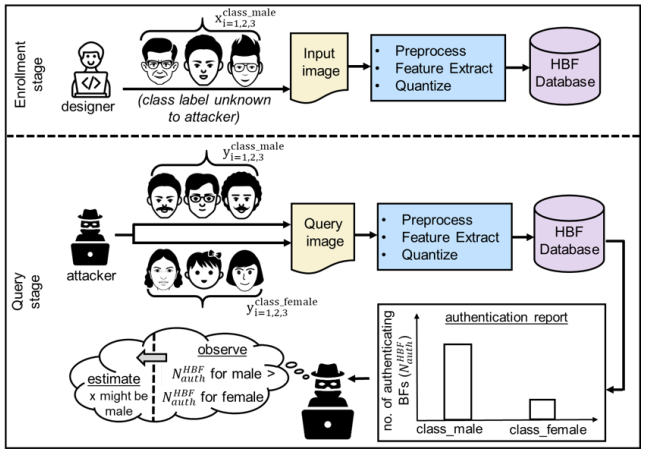

Fig. 9: Query attack for information leakage about enrolled individuals, i.e., attacker collects information by assessing HBF's authentication decision).

the gender, age, and ethnicity information as the features of the enrolled users which the attackers want to leak from HBF. As an example, Figure 9 shows how an attacker tries to leak the gender information about the enrolled users by providing query samples from male and female classes and assessing the HBF's decisions about them.

Gender leakage: There are two classes, namely, male and female for this experiment. In the HBF for UTKface, only the $8,100$ samples who are male were enrolled. For, $n = 8,100$, we obtain, $m = 38,820$ and $k = 4$ from formulas provided in [32]. As query sets, $1,800$ unenrolled templates from both male and female classes are provided. As shown in Figure 10(a), query samples from both of the classes obtain $N_{auth} \approx 6$ BFs on average. If the HBF was gender-biased, it would have authenticated the male query samples with a higher number of BFs, eventually resulting in more male sample authentication. Since this has not happened, it can be inferred that HBF is not biased towards the authentication for the query samples who share the same feature (gender as a feature here) with enrolled templates, and thus the attacker can never reveal about the enrolled users' gender information from HBF. Note that the same experiment was performed for all females enrolled and different proportions of males and females enrolled. The results were similar and not shown for brevity.

Ethnicity leakage: A similar experiment has been repeated to see whether the HBF leaks ethnicity information. With an HBF with only $n = 7,200$ white people enrolled, the query set of $4,000$ individuals provided to the HBF in consisting of unenrolled samples from five different classes, namely white, black, Asian, Indian, and other. The results from Figure 10(b) show that no matter what is the ethnicity of the query templates, they all get treated similarly by the BFs in HBF and authenticated by almost the same number of BFs. which is around 6. Once again, regardless of the enrolled class, the result was the same.

Age leakage: For this experiment, samples from five different groups are considered: age 1 to 15 years, 16 to 29 years, 30 to 49 years, 50 to 70 years, and 71 to 116 years. Note that, this split is inspired by the authors in [2]. As an enrollment set, only $n = 5400$ samples from the 16 to 29 years age group are considered. However, when $3,000$ unenrolled samples from all 5 age groups are presented to the HBF, irrespective of their age, around $N_{auth} \approx 6$ BFs, in general, authenticate them. Thus, it does not matter if a query sample has the same age as the enrolled users. The HBF is not biased by such features when authenticating a query which ultimately implies that the attacker can never get age-related information from HBF about its enrolled users. This same experiment was repeated with enrolled groups from other ages with the same result.

Bias check with wolves: An experiment was also conducted to see whether the authentication performance of an HBF is biased by the wolves or not. According to [10], wolves are the imposters who can legitimately impersonate the enrolled users because of their high similarity with them. We conducted an experiment where the enrolled items are males only. As the query templates, two types of templates were provided. In the type 1 query template, there are only wolves of the enrolled templates who are wolves (male and have a similarity score with the enrolled templates > 0.65). As the type 2 query set, we provide some female samples and male samples who are not wolves of the enrolled items (similarity score < 0.65). As shown in Figure 11, it appears that for both of the cases, only around 8 BFs authenticated the samples of each type of query set which means the HBF decision is not biased by wolves either.

B. Exploiting BF_i Distribution for Information Leakage

The second set of experiments have been designed to scrutinize the role of these authenticating BFs individually to have a more thorough view of the HBF's security measures. The objective of these experiments is to verify whether the authenticating BFs are chosen uniformly while authenticating the query samples from the same class. An illustration is shown in Figure 12, where the attacker submits query samples for male and female class and since she finds that BF_1 has a tendency to authenticate female samples more than male samples, she can guess that there must be some female individuals who are enrolled.

These have been conducted in two ways to check the individual BF's role (if any) in leakage. First, a statistical hypothesis test is performed to check the distribution of the authenticating BFs. If the distribution is uniform, it implies that the choice of authenticating BF is random; therefore the attacker cannot link the individual BF's role to the samples of a particular class. Second, classifiers were trained to see if the classification was possible among the query data utilizing the authentication results provided by HBF. The second experiment is just an illustration of the first experiment, i.e., if the choice of the BF in the HBF follows uniform distribution every time while authenticating query samples from the same class, using this information, it is not possible to classify among the query samples and thus relate them to a particular BF. For both the experiments, gender, ethnicity, and age information are once again considered as soft information. Also, note that, unlike the experiment in Section VI-A, the current experiment

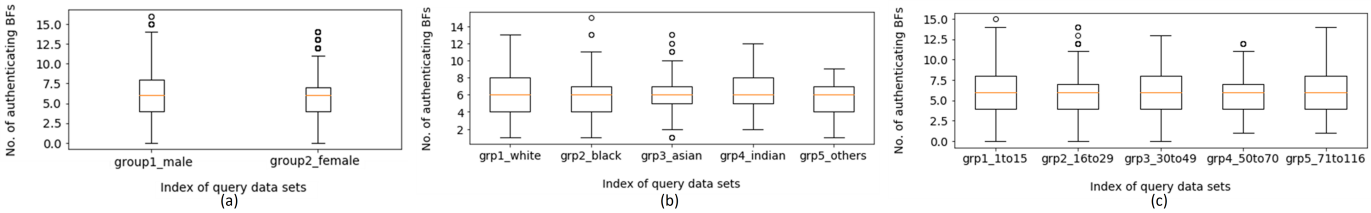

Fig. 10: HBF bias towards enrolled samples for (a) gender, (b) ethnicity, and (c) age.

Fig. 11: HBF bias towards wolves.

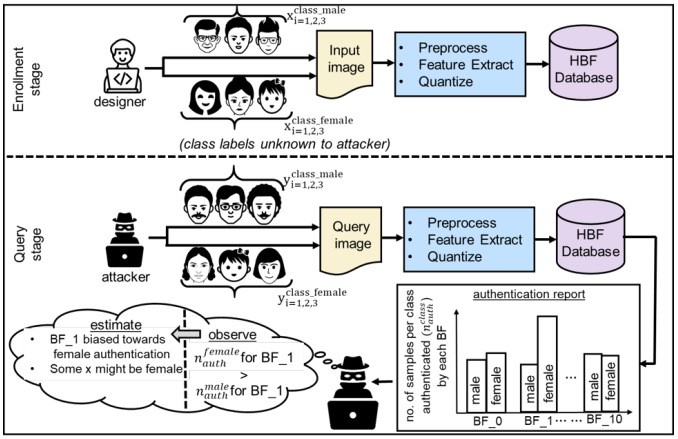

Fig. 12: Query attack for information leakage about enrolled individuals, i.e., attacker collects information by assessing individual BF authentication decisions.

requires the enrollment of at least few samples from all the classes. The experiment details are provided below.

1) Statistical hypothesis test for analyzing the authenticating BFs: In our next set of experiments, we perform the Chi-square goodness-of-fit test as a statistical hypothesis test. For this purpose, a null hypothesis H_0 is defined as "the data is distributed uniformly." For such a test, two sets of data can be considered, where one follows an expected distribution and another is observed. According to the hypotheses, it is assumed that the distribution of observed data follows the theoretical (= expected) distribution. A test statistic is calculated to measure how far the observed data are from the null expectation. In our case, we then use the Chi-square distribution to estimate the probability of obtaining that value of the test statistic. Based on this and a predefined significance level (0.05 in our experiments), the hypothesis is rejected or accepted. In the latter case, the observed data set follows the distribution considered in the hypothesis, i.e., the uniform distribution in our experiments.

Moreover, as in some cases, we have to deal with more than two categories, and some of the categories might have small numbers, we employ the pooling technique. This technique pools some of the categories together to answer the question related to the H_0 hypothesis. In our experiments, the question that we are interested in is: are the cell indices corresponding to inputs of the same class distributed uniformly inside BFs? The results of our experiments, as described below, positively answer this question.

Gender-based classes: For this experiment, the enrollment set consists of $n = 10,000$ samples consisting of $5,000$ male and $5,000$ female samples. After the enrollment is done, $1,000$ query samples from both male and female classes are provided. It is observed that on average $N_{auth} = 6$ BFs are authenticating query samples for each class (see Figure 13(a)). Now, the objective of this experiment is to analyze each of the N_{auth} BF's role to examine if these BFs are the same for the different query samples from the same class. To verify this, a simple way is to determine the distribution of these BFs. When a male sample is queried from the HBF, the N_{auth} BFs authenticating it should ideally be different from the BFs authenticating another male sample. If the same BFs are authenticating most of the query samples, that means the BFs individually are biased towards the gender of these samples. On the other hand, if the cells in N_{auth} BFs do follow a uniform distribution, this implies that different BFs authenticate the male samples. Therefore, the attacker cannot relate a BF (or a sub-set of BFs) to a specific class (e.g., male here). To check whether the BFs follow uniform distribution or not, we perform hypothesis testing.

In this experiment, we considered two pools. The first pool draws the BF index values $\in [0, N-1]$ of the BFs as "observed class" that is responsible for authenticating the query samples from the male class. The other pool consists of cells that are not set by the entries from the observed class. The number of samples in each pool is $6,948$. After performing the test, it is observed that the Chi-square test passes with a 5% level of significance (i.e., 5% risk of incorrectly rejecting the null hypothesis). The same test has been repeated for the female class as well where the number of samples in the pools is $6,886$ and the test also passed with a 5% level of significance.

Ethnicity-based classes: For this experiment, the enrollment set consists of five different classes of 3,000 white, 2000 black, 2,000 Asian, 2,000 Indian, and 1,000 other ethnicity-based samples. Thus, $n = 10,000$. Unlike the gender-based experiment, an equal number of samples in each class is not provided in the dataset. After the enrollment step, the query processing is performed with $1,000$ query samples from the

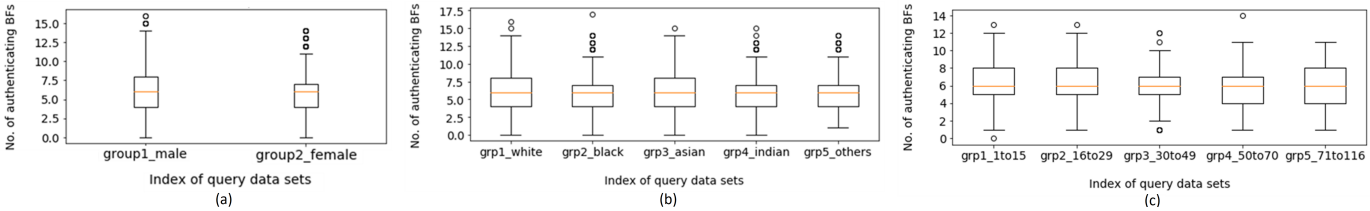

Fig. 13: HBF authentication performance when samples from each class is enrolled for (a) gender, (b) ethnicity, and (c) age.

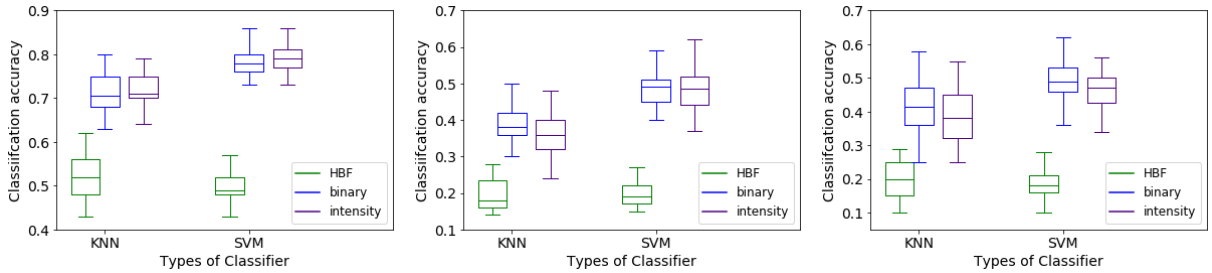

Fig. 14: Classification performance on three different feature sets (HBF, binary, and intensity) for 3 types of soft information: (a) gender (b) ethnicity, and (c) age. Note that, the three different features sets "HBF", "binary", and "intensity" denote individual BF's decision derived features, distribution-based binary feature, and intensity-based features, respectively. The binary and intensity-based features are the input to the HBF and show correlation with soft information whereas the HBF is uncorrelated.

five individual classes. An important observation here similar to the experiments on Section VI-A is that, despite having unequal enrollment samples, the average N_{auth} for samples for each class is almost equal to 6 (see Figure 13(b)). This again supports the overall privacy of the enrolled users in terms of ethnicity. However, getting back to the objective of the current experiment, we are interested in finding out whether the distribution of these N_{auth} follows uniform distribution while authenticating query samples from the same class. The Chi-square test was performed to find the distribution of each of N_{auth} BFs for five different classes. The number of items in each pool is $7,770$, $6,682$, $7,948$, $6,964$, and $6,904$, respectively. The Chi-square tests successfully passed in all five cases with the level of significance set to 5%.

Age-based classes: This experiment involves $n = 9,000$ samples from five different classes as enrollment sets, consisting of $2,270$ samples from the first age group, $2,270$ samples from the second, $2,270$ samples from the third, $2,270$ samples from the fourth, and 920 samples from the fifth. The reason for inequality in the number of samples in a different class is regarding the availability of samples per class in the dataset. On the other hand, the query set involves $1,000$ samples from the five classes. After the query processing step, it is observed from Figure 13(c)) that the average N_{auth} for the query samples from all five classes is ≈ 6, which supports the observation from Section VI-A – the N_{auth} BFs act independently of the features (age here) of enrollment data. Our Chi-square tests on the distribution of BFs for five different classes reveal that the BFs have followed a uniform distribution each with the level of significance set to 5%.

2) Classification test for analyzing the authenticating BFs: Another method to check whether a particular BF tends to authenticate samples in a particular class more than that of another class is performing a classification experiment on the query samples is based on each BF's authentication decisions. To illustrate, once the query processing is done, for each of the query samples a new template can be generated which will reflect each of the BF's authentication decision during querying that sample. If performing a classification task on these newly produced templates shows an efficient classification accuracy for all the classes, this will imply that the authentication performance of each BF towards the different samples from the same class is alike. This results in a follow-up observation in which the attacker might be interested, which is, if maximum samples of a particular class are getting authenticated by certain BFs, the HBF may have enrolled the users who have similar features as those query samples. Thus, this leads to information leakage about the enrolled users. For better security measures and a low/no information leakage environment, poor classification accuracy is desired for HBF. The experiments below on gender, ethnicity, and age-based feature are again considered for the classification experiments.

Gender-based classes: Based on the authentication decision of the individual BF towards a query sample, a new template will be generated for the query sample. As there are N number of BFs in the HBF, then the new query template will be an N bit long binary template where each index in the template represents the authentication decision ('0' for rejected and '1' for authenticated) by each BF in the HBF. This binary template generation procedure is repeated for all the query templates under all the classes. When the new binary templates are ready, a classifier is trained.

In our gender experiment since there are $2,000$ query samples ($1,000$ samples in each class), a new set of query

template-based on HBF's authentication decision is produced with $2,000$ binary templates. This set is now considered as the new input dataset for the classification experiment from which a training and testing set is produced with 0.2 as a train-test-split ratio. Considering two different classifiers, namely, KNN and SVM, the classification accuracy is obtained as 52% and 50% on average. Hence, the information surprisal metric can measure the loss of privacy as $priv_{IS,KNN} = 0.94$ and $priv_{IS,SVM} = 1$, where $priv_{IS} = 1$ indicates the maximum privacy (see equation (5)). As the loss of privacy values are in close proximity with the maximum value, it can be assumed that the HBF preserves privacy almost perfectly. Moreover, the percentage incorrectly classified is calculated: $priv_{PIC} = 50\%$. This shows that the adversary cannot guess with a probability better than tossing a coin, and consequently, privacy is not lost based on this metric.

The implication of these results is that the poor accuracy value denotes restricted information leakage as well as better privacy provided by the HBF framework. To illustrate, since the classification performance is poor, the BFs do not tend to consider the samples with the same gender as a class and therefore, do not impose a similar kind of authentication decision on them. In other words, if there are samples from 2 different classes, if one sample is chosen randomly, the probability of correctly guessing its class would be 1/2. Here in Figure 14a, for HBF, the results reveal that, the individual BFs show authentication possibilities 50-50% for a 2 class system (male and female). Thus, based on each of the BF's authentication decisions, it is not possible to classify or identify if a BF is authentication more male samples or female samples. Since this information is unidentifiable, the attackers cannot guess the gender of the enrolled users in the HBF by assessing the individual BF's authentication performance.

On the other hand, two different feature representation methods are considered to verify if the classifiers are biased towards their gender information or not in non-BF-based frameworks. The non-BF-based features are: distribution-based binary features and intensity-based features. As shown in Figure 14a, both of the feature set show higher classification accuracy ($\approx 72\%$ for KNN and for $\approx 80\%$ SVM) than that of the HBF-based features. This simply implies that HBF does not get biased with the gender information of the query templates. That's the reason why the classifiers poorly classified the HBF's decision derived query templates. However, for the other two feature representations, they are biased by the gender information, thus the classifiers performed better on them.

Ethnicity-based classes: The classification performance is also verified with ethnicity-based classes where $5,000$ samples from the five classes are generated. As shown in Figure 14b, for HBF, the classification accuracy for KNN is $\approx$ 18%, whereas for SVM it is $\approx$ 19%. Note that, if there are samples from 5 different class and one sample is randomly picked, the probability of guessing its class = 20%. In this experiment, $priv_{IS,KNN} = 2.47$ and $priv_{IS,SVM} = 2.39$, where $priv_{IS} = 2.32$ demonstrate the maximum privacy (see Equation (5)). Furthermore, the percentage incorrectly classified is calculated: $priv_{PIC} = 81\%$. This again demonstrates that the BFs in the HBF are not impacted by the ethnicity information of the samples while authenticating them. On the other hand, the classifiers for both of the non-BF-based features show higher classification accuracy ($\approx 40\%$ for KNN and $\approx 50\%$ for SVM) (see Figure 14b).

Age-based classes: The classification experiment is repeated for age-based classes too where $5,000$ samples from each group. As shown in Figure 14c, the classification accuracy for KNN and SVM is $\approx$ 18% and 20% respectively (which is similar to guessing the class of a randomly picked item from five different class). Similar to the experiments performed on gender and ethnicity classes, we compute $priv_{IS,KNN} = 2.47$ and $priv_{IS,SVM} = 2.32$, where $priv_{IS} = 2.32$ demonstrate the maximum privacy (see Equation (5)). In addition to this, the percentage incorrectly classified is calculated: $priv_{PIC} = 81\%$. Thus it can be inferred that the BFs in the HBF are not impacted by the age information of the sample while authenticating, and does uniformly treat them. As a result, that attacker cannot relate the individual BF's authentication decision to the provided query data as well as guess the ethnicity of the enrolled users. On the other hand, the classifiers show higher classification accuracy ($\approx 40\%$ for KNN and $\approx 49\%$ for SVM) for both of the non-BF-based features (see Figure 14c), meaning the classifiers rely on the age-based feature while doing classification and thus, the attacker can also guess the age of the query templates by observing these classification result patterns.

VII. Related Work

As mentioned in Section I, the four major challenges associated with any biometric system implementation include a reduction in storage requirement and query processing time, handling noise in the template, and ensuring the privacy of the user data. Numerous biometric systems around us are deployed for diverse applications; however, none of them address all the challenges simultaneously. While some studies have focused on strengthening the security of the biometric system, unlike HBF, they do not provide all the other desirable features (i.e., storage and search-time efficiency, template security, noise resistance) simultaneously. Here, we classify existing biometric systems into two groups, non-NF-based and BF-based, present the security aspects of the related studies on biometric authentication systems, and delineate how HBF's security arrangement is unique.

A. Non-BF-based Biometric System

Being very fast for even large databases, nearest-neighbour (NN) search algorithms are popular for biometric applications [3, 16, 28]. Due to the high demand for privacy-preserving features in the database, modern NN techniques emphasize adopting encrypted databases to protect the biometric templates of the users [22, 26, 35]. However, two recent studies have demonstrated that data recovery attacks in encrypted databases can be possible with judiciously chosen queries [19, 20]. Thus, the privacy-preserving encrypted databases used in modern NNs are prone to information leakage attacks. Another popular NN search technique is locality sensitive

TABLE II: Comparison between our proposed solution and related works

Use of BF	Non-BF-based					BF-based			
Approach	Nearest neighbor search					BF for iris	BF for face	BF for fingerprint	Proposed
Ref	[3]	[28]	[35]	[22]	[26], [29], [17]	[23], [24]	[12], [25]	[1], [21]	[31]
Orig. template unrequired	×	✓	✓	✓	✓	✓			✓
Stores modified templates	×	✓	✓	✓	✓	✓			✓
Resists info. leakage	×	×	×	×	×	×			✓
In house template modification	–	×	×	×	✓	✓			✓
Template protection technique	–	Homomorphic enc., garbled circuit	Enc.	Enc., hash	Hash	Bijective numeration			Hash

hash-based matching which is very popular because of its high classification performance for high dimensional data [17, 29]. However Riazi et. al. has demonstrated that LSH-based systems can be prone to triangulation attack because the LSH-based binary embedding in the system store all the pairwise distances between the samples which contrarily reveal enough sensitive information about the samples, thus making user identities vulnerable [26].

B. BF-based Biometric System

Recently, the BF-based biometric system development has gained much popularity due to three major features [13]. Being a probabilistic data structure, the BF uses a very compressed database (DB) to replace a large DB and offers very fast membership query preprocessing. The BF-based data structure also protects the underlying biometric information of the users. Moreover, the structure can be well adapted for diverse biometric traits as well, e.g., [23, 24] considered BF for iris, [12, 25] face, and [1, 21] fingerprints. However, the major drawback of these BF-based biometric systems is that the studies avoided the inclusion of cryptographic hash functions as originally advised by the theory of BF [32]. As a rationale, the authors at [23, 24, 25] assumed their data to be collision-free, hence considered a simple binary to decimal conversion as "hash" and the same steps were later followed in [1, 13, 21]. However, in reality, biometric data is noisy by nature and due to the noise, two templates from the different individuals can easily collide. Therefore, the assumption about the collision-free nature of data does not hold in practical scenarios. Further, due to the avoidance of the proper hash function the privacy and security features of the existing systems could be at risk. The underlying security risk at [24] has also been explored by authors at [14]. With intensive experiments, they have shown that the weak security is due to the consideration of a single and simple hash function. There exist some additional studies which consider BF however seem to be unjustified for comparison with HBF. For instance, the authors at [36] present a BF based approach for preserving the privacy of input samples. There are three major differences for which [36] cannot be compared to our proposed work. First, [36] is privacy preserving based machine learning technique whereas the HBF is an access control system. Second, [36] is applicable for sequential data whereas ours is not applicable for sequential data. Third, the authors of [36] consider the BF as a binary feature extraction technique and they incorporate multiple concepts (e.g., dimensionality reduction, data perturbation, order information addition, custom noise addition, etc.) to ensure privacy in the extracted features. On the contrary, our proposed HBF is not used for extracting features from templates, rather it is only enrolling them for allowing membership query processing in the query stage. Thus, it is not justified to compare HBF with the method suggested in [36].

The authors of [5] have assessed the security of spatial BFs (a variant of BF) and recommended it for number of secured practical application. However, a comparison between the legacy SBF and proposed HBF is not reasonable because the attacker for SBF is interested in inter-set FP authentication (FP between different databases) whereas the proposed approach is concerned about intra-set error (FP between members under the same database). Moreover, the authors only mention that SBF can be useful for biometric template protection application, however, implementing the legacy SBF for biometric database is challenging because of the fuzziness of the data. Thus, the data structure will require some modification for incorporating biometric fuzziness and then it will be justified to get compared to HBF. Finally, the authors at [30] propose a modified BF scheme for biometric template protection by incorporating a key for encoding the raw templates before inserting into BF. Since the study does not mention about handling fuzziness and besides has a dependence on the encoding and decoding schemes for ensuring privacy, the comparison between [30] and HBF seems invalid.

According to [11], the cryptographic hash function acts as a strong one-way hash function. Therefore, when applied in BF-based biometric systems, they offer irreversible transformation of the biometric templates during enrollment [11], enhancing the security of the biometric system. However, the application of hash functions on the templates during enrollment can cause complexity during authentication. Therefore, we proposed the HBF-based biometric system in [31] which is the first work where the hash functions are applied as suggested in theory [32] and the noise handling property is enhanced by introducing a hierarchical framework of multiple BFs. In [31], the details about the construction of such a system and formulas for choosing the optimum values of parameters are provided. These factors are very crucial because if designed and chosen properly, the overall system including the cryptography function in the HBF-based biometric system prevents the attackers from collecting sensitive and confidential information about the enrolled user and at the same time prevents manipulation

of authentication output. Thus, as shown in Table II, the HBF-based biometric system is superior to the state-of-the-art contemporary solutions.

VIII. Conclusion

Probabilistic data structures, such as HBF, are promising candidates for modeling biometric systems because of the fast query processing and storage efficiency properties, but their security and privacy features require thorough evaluation before they gain wider acceptance. In this paper, we defined attack models for an HBF-based biometric system and analyzed them qualitatively and quantitatively to determine if it can withstand attacks. Even while considering the most challenging scenario where the construction of HBF and the user enrolled HBF is public, our experimental results and analytical expressions have shown that the HBF does not reveal any sensitive or confidential information about the enrolled users and the authentication results cannot be manipulated at reasonable cost. The latter is even more difficult than a classical BF.

Acknowledgment

The authors would like to thank US Army Research Office for granting award W911NF-19-1-0102 to support the study.

References

[1] N. Abe, S. Yamada, and T. Shinzaki, "Irreversible fingerprint template using minutiae relation code with bloom filter," in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2015, pp. 1–7.

[2] V. Albiero, K. Bowyer, K. Vangara, and M. King, "Does face recognition accuracy get better with age? deep face matchers say no," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 261–269.

[3] H. binti Jaafar, N. binti Mukahar, and D. A. B. Ramli, "A methodology of nearest neighbor: Design and comparison of biometric image database," in *2016 IEEE Student Conference on Research and Development (SCOReD)*. IEEE, 2016, pp. 1–6.

[4] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[5] L. Calderoni, P. Palmieri, and D. Maio, "Probabilistic properties of the spatial bloom filters and their relevance to cryptographic protocols," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1710–1721, 2018.

[6] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 804–815, 2015.

[7] T. Chen, A. Chaabane, P. U. Tournoux, M.-A. Kaafar, and R. Boreli, "How much is too much? leveraging ads audience estimation to evaluate public profile uniqueness," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2013, pp. 225–244.

[8] D. Clayton, C. Patton, and T. Shrimpton, "Probabilistic data structures in adversarial environments," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1317–1334.

[9] B. Coskun and N. Memon, "Confusion/diffusion capabilities of some robust hash functions," in *2006 40th Annual Conference on Information Sciences and Systems*. IEEE, 2006, pp. 1188–1193.

[10] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds, "Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation," National Inst of Standards and Technology Gaithersburg Md, Tech. Rep., 1998.

[11] T. Gerbet, A. Kumar, and C. Lauradoux, "The power of evil choices in bloom filters," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2015, pp. 101–112.

[12] M. Gomez-Barrero, C. Rathgeb, J. Galbally, J. Fierrez, and C. Busch, "Protected facial biometric templates based on local gabor patterns and adaptive bloom filters," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 4483–4488.

[13] M. Gomez-Barrero, C. Rathgeb, G. Li, R. Ramachandra, J. Galbally, and C. Busch, "Multi-biometric template protection based on bloom filters," *Information Fusion*, vol. 42, pp. 37–50, 2018.

[14] J. Hermans, B. Mennink, and R. Peeters, "When a bloom filter is a doom filter: security assessment of a novel iris biometric template protection system," in *2014 international conference of the biometrics special interest group (BIOSIG)*. IEEE, 2014, pp. 1–6.

[15] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Journal on advances in signal processing*, vol. 2008, p. 113, 2008.

[16] U. Jayaraman and P. Gupta, "Efficient similarity search on multidimensional space of biometric databases," *Neurocomputing*, 2021.

[17] Z. Jin, J. Y. Hwang, Y.-L. Lai, S. Kim, and A. B. J. Teoh, "Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 393–407, 2017.

[18] K. Kärkkäinen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age," *arXiv preprint arXiv:1908.04913*, 2019.

[19] G. Kellaris, G. Kollios, K. Nissim, and A. O'neill, "Generic attacks on secure outsourced databases," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1329–1340.

[20] M.-S. Lacharité, B. Minaud, and K. G. Paterson, "Improved reconstruction attacks on encrypted data using range query leakage," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 297–314.

[21] G. Li, B. Yang, C. Rathgeb, and C. Busch, "Towards generating protected fingerprint templates based on bloom filters," in *3rd International workshop on biometrics and forensics (IWBF 2015)*. IEEE, 2015, pp. 1–6.

[22] Y. Peng, H. Li, J. Cui, J. Ma, and Y. Liu, "Towards secure approximate k-nearest neighbor query over encrypted high-dimensional data," *IEEE Access*, vol. 6, pp. 23 137–23 151, 2018.

[23] C. Rathgeb, F. Breitinger, H. Baier, and C. Busch, "Towards bloom filter-based indexing of iris biometric data," in *2015 international conference on biometrics (ICB)*. IEEE, 2015, pp. 422–429.

[24] C. Rathgeb, F. Breitinger, C. Busch, and H. Baier, "On application of bloom filters to iris biometrics," *IET Biometrics*, vol. 3, no. 4, pp. 207–218, 2014.

[25] C. Rathgeb, M. Gomez-Barrero, C. Busch, J. Galbally, and J. Fierrez, "Towards cancelable multi-biometrics based on bloom filters: a case study on feature level fusion of face and iris," in *3rd international workshop on biometrics and forensics (IWBF 2015)*. IEEE, 2015, pp. 1–6.

[26] M. S. Riazi, B. Chen, A. Shrivastava, D. Wallach, and F. Koushanfar, "Sub-linear privacy-preserving near-neighbor search," *arXiv preprint arXiv:1612.01835*, 2016.

[27] K. Ricanek and T. Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. IEEE, 2006, pp. 341–345.

[28] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *International Conference on Information Security and Cryptology*. Springer, 2009, pp. 229–244.
[29] D. Sadhya, Z. Akhtar, and D. Dasgupta, "A locality sensitive hashing based approach for generating cancelable fingerprints templates," in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2019, pp. 1–9.
[30] D. Sadhya and S. K. Singh, "Providing robust security measures to bloom filter based biometric template protection schemes," *Computers & Security*, vol. 67, pp. 59–72, 2017.
[31] S. Shomaji, F. Ganji, D. Woodard, and D. Forte, "Hierarchical bloom filter framework for security, space-efficiency, and rapid query handling in biometric systems," in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2019, pp. 1–8.
[32] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and practice of bfs for distributed systems," *IEEE Comm. Surveys & Tutorials*, vol. 14, no. 1, pp. 131–155, 2012.
[33] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–38, 2018.
[34] Z. Wang, X. Tang, W. Luo, and S. Gao, "Face aging with identity-preserved conditional generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7939–7947.
[35] R. Xu, K. Morozov, Y. Yang, J. Zhou, and T. Takagi, "Efficient outsourcing of secure k-nearest neighbour query over encrypted database," *Computers & Security*, vol. 69, pp. 65–83, 2017.
[36] W. Xue, D. Vatsalan, W. Hu, and A. Seneviratne, "Sequence data matching and beyond: New privacy-preserving primitives based on bloom filters," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2973–2987, 2020.
[37] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5810–5818.

Sumaiya Shomaji received the B.S. degree in electrical and electronics engineering from the Ahsanullah University of Science and Technology, Dhaka, Bangladesh in 2013. She worked as a lecturer at Stamford University Bangladesh from January 2014 to May 2015. She received her M.S. and Ph.D. degrees from the University of Florida, Gainesville, FL in 2021. Her research interests include application of machine learning, image processing, computer vision, and blockchain to solve emerging challenges in biometrics, hardware security, internet of things, and healthcare. She will join the Computer Science Department at Texas Tech University as an Assistant Professor from Spring 2022.

Pallabi Ghosh Pallabi Ghosh received the B.Tech degree in Information Technology from the Institute of Engineering and Management, Kolkata, India in 2015, the M.S. degree in Computer Science and Engineering from the Indian Institute of Technology (IIT), Kharagpur (India) in 2019. She is currently pursuing Ph.D. in the Department of Electrical and Computer Engineering at the University of Florida. She worked as a Junior Research Fellow in the Secured Embedded Architecture Laboratory (SEAL) of IIT Kharagpur during her M.S. She is currently working as a Graduate Research Assistant in the Florida Institute for Cybersecurity (FICS) Research. Her research interests include computer vision, deep learning, machine learning and biometric.

Fatemeh Ganji received the Ph.D. degree in electrical engineering from the Technical University of Berlin, in 2017. She was a Postdoctoral Research Fellow with T-Labs, Telekom Innovation Laboratories, the Technical University of Berlin, and Postdoctoral Fellow with the Florida Institute for Cybersecurity (FICS) Research, University of Florida. She is currently an Assistant Professor with the Electrical and Computer Engineering Department, Worcester Polytechnic Institute, Worcester, MA, USA. She has focused her research activities on the applied and theoretical machine learning techniques and mathematical tools for the security assessment of hardware primitives, for instance, physically unclonable functions (PUFs). For her work on the learnability of PUFs, she received the BIMoS Ph.D. Award, in 2018.

Damon Woodard received the B.S. degree in computer science and computer information systems from Tulane University, in 1997, the M.E. degree in computer science and engineering from Pennsylvania State University, in 1997, and the Ph.D. in computer science and engineering from the University of Notre Dame, in 2005. From 2006 to 2014, he was an Assistant Professor and then an Associate Professor with the School of Computing, Clemson University, Clemson, SC, USA. In 2015, he joined the Faculty of the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA, as an Associate Professor. He is also a member of the Florida Institute for Cybersecurity (FICS) Research. His research interests include biometrics, machine learning, pattern recognition, natural language processing, and signal/image analysis. Dr. Woodard is an ACM Senior Member and the National Academy of Science Kavli Frontier Fellow.

Domenic Forte received the B.S. degree from the Manhattan College, Riverdale, NY, USA, in 2006, and the M.S. and Ph.D. degrees from the University of Maryland at College Park, College Park, MD, USA, in 2010 and 2013, respectively, all in electrical engineering. He is currently an Associate Professor and the Steven A. Yatauro Faculty Fellow with the Electrical and Computer Engineering Department, University of Florida, Gainesville, FL, USA. His research interests include the domain of hardware security, including the investigation of hardware security primitives, hardware Trojan detection and prevention, electronics supply chain security, and anti-reverse engineering. He was a recipient of the Presidential Early Career Award for Scientists and Engineers (PECASE), the Early Career Award for Scientists and Engineers (ECASE) by the Army Research Office (ARO), the NSF Faculty Early Career Development Program (CAREER) Award, and the ARO Young Investigator Award. His research has also been recognized through multiple best paper awards and nominations.