SimplePIR on Biometric Authentication

Zhiron Wu (zwuy@mit.edu)

Robin Xiong (rxiong22@mit.edu)

Catherine Zhu (cathzhu@mit.edu)

Abstract

As digital identity becomes crucial in modern systems, the transition from knowledge-based methods to biometric authentication has become increasingly common. However, the immutability of biological features introduces unique privacy risks. Unlike traditional passwords, biometrics are unchangeable, creating opportunities for malicious parties to track individuals across systems if authentication data or metadata is leaked. This paper tests the feasibility of integrating Private Information Retrieval (PIR) into biometric authentication pipelines to prevent metadata leakage. We focus on an "honest-but-curious" threat model, commonly seen in outsourced third-party servers.

We implement and evaluate two distinct approaches, Direct PIR and Bucketed PIR. Both approaches are implemented using SimplePIR, a single-server protocol based on the Learning with Errors (LWE) assumption. To reduce PIR's computational overhead, we use a pretrained FaceNet model to extract 512-dimension embeddings from the Labeled Faces in the Wild (LFW) dataset and apply 8-bit integer quantization to reduce the template size.

Our results show that Direct PIR is a scalable option for medium-scale uses. On the other hand, our Bucketed PIR approach that uses Locality Sensitive Hashing (LSH) to reduce the search space for PIR resulted in significant latency. While it is theoretically more efficient, we conclude that Bucketed PIR requires further refinement to become practical.

Keywords: Private Information Retrieval, Biometric Authentication, SimplePIR, Locality Sensitive Hashing, FaceNet

Introduction

As processes today become increasingly digitalized and digital identity becomes central to global infrastructure, the methods used to verify identity are evolving to become faster and more accurate. In particular, a shift from knowledge-based systems (like passwords) to biological systems is prevalent due to their ability to be fast and generally fraud-resistant.

Specifically, **biometric authentication**, a method of authentication using unique physical features such as facial features or fingerprints, is becoming increasingly common. However, this transition creates significant privacy concerns because, unlike traditional passwords which can be changed if compromised, biometric data is immutable and remains constant. This makes the privacy of biometric authentication a necessity [1].

Privacy Concerns With Remote Authentication

In many large-scale applications, it is often infeasible to store millions of biometric templates locally due to hardware constraints [2]. As a result, authentication may be outsourced to third-party servers.

While these servers are generally reliable, they may be **"honest-but-curious."** Current authentication inherently leaks sensitive metadata surrounding which individuals are attempting to authenticate and when. With unchangeable biometric features, compromising authentication data allows individuals to be tracked across different systems and locations.

A primary example of an application that possesses these security concerns is facial-recognition kiosks at international airports, where millions of people travel through each day. Storing such a large number of biometric templates on local hardware is difficult, requiring a centralized database. This authentication metadata is especially sensitive, since it contains their location and travel timing. This data makes it easy for an honest-but-curious server to monitor behavioral patterns and track individuals around the world, creating a huge surveillance hazard.

Private Information Retrieval

To address the privacy concerns surrounding identity leakage, we investigate **Private Information Retrieval (PIR)**. PIR, first introduced by Chor et al. [3], is a cryptographic protocol that allows a client to retrieve an entry from a database without the server learning which specific index was requested. Early iterations of PIR required multiple non-colluding servers to achieve privacy, but advancements by Kushilevitz and Ostrovsky have enabled PIR for single server applications [4].

Currently, PIR is commonly used in sensitive lookups involving medical database queries, DNS resolutions, and credential checking for compromised passwords. Despite its successful performance in a variety of use cases, it has limited appli-

cation in authentication. This is largely due to its significant communication and computation overhead, which often clashes with the speed requirements of modern authentication systems.

Objectives and Contribution

In this paper, we argue that the unique risks that come with biometric data, specifically its immutability and therefore value to adversaries, justify the overhead of PIR more than traditional password-based systems. We explore the feasibility of integrating PIR into biometric authentication to allow a server to return a template without knowing which user is authenticating.

We evaluate the performance and feasibility of two distinct PIR approaches: **Direct PIR** and **Bucketed PIR**.

Related Work

Privacy-Preserving Biometric Authentication

The deployment of biometric systems in high-throughput identification settings, such as our motivating scenario of border control and airport security kiosks, introduces a delicate balance between utility and privacy. In these environments, clients cannot store a full biometric database locally, making a centralized server necessary, which introduces a point of vulnerability for metadata leakage at the same time. Even with encrypted templates, an honest-but-curious server can learn which individual's record is being queried simply by observing access patterns. Early work by Erkin et al. [5] addressed this by proposing one of the first strongly privacy-enhanced face recognition systems, utilizing secure Multi-Party Computation (MPC) to run standard eigen-face recognition algorithms. In the domain of fingerprints, Kerschbaum et al. [6] described a protocol for comparing fingerprints without exchanging them, and Shahandashti et al. [7] later proposed a fully private minutia matching protocol secure against honest-but-curious adversaries. While these works establish the baseline privacy requirements for our project—specifically, that the server should learn neither the fresh facial scan nor the retrieved template, their reliance on computationally heavy algorithms or multi-round MPC makes them difficult to scale in fast-paced kiosk environments. Our work addresses this by shifting the computational paradigm to single-server PIR to minimize interaction rounds.

PIR Applied to Biometric Authentication

Integrating PIR into biometric protocols addresses the access-pattern leakage problem that encryption alone cannot solve. Bringer, Chabanne, Pointcheval, and Tang [8] introduced Extended PIR (EPIR), which evaluated a function over a client's string and a server-held database block. Their protocol effectively computed a Hamming distance without either party learning the other's input, proving that EPIR could be viable for privacy-preserving authentication. In 2008, Bringer and Chabanne [9] improved this by replacing the underlying PIR scheme with a more communication-efficient construction by Lipmaa. However, these foundational works primarily targeted biometric templates that could be evaluated using simple Hamming distances over binary strings, such as iris codes. Adapting these cryptographic principles to modern facial recognition presents a fundamental mathematical challenge: PIR schemes operate exclusively over finite integer fields, while deep learning models like FaceNet output continuous floating-point vectors. We bridge this gap through quantization. By mapping the 512-dimensional continuous embeddings into discrete 8-bit integers, we can securely evaluate complex deep-learning templates within the strict mathematical bounds of PIR, ultimately using cosine similarity for the final authentication threshold.

Direct vs Bucketed PIR

A central limitation of standard Private Information Retrieval, known as Direct PIR throughout this paper, is that the query space encompasses the entire database. To maintain cryptographic blindness, the server must perform a computation across every single record in the database during each retrieval. As surveyed by Ostrovsky and Skeith [10], this requires a linear scan where computational costs scale directly with the size of the database (N). In our Direct PIR implementation using SimplePIR, this follows as $O(N)$ matrix operations, which can become a significant bottleneck when there are more users in the database. Bucketed approaches address this by reducing the effective database size before invoking PIR. Rather than querying over all templates, the client first computes a bucket index from its own biometric data, comparing it to planes to retrieve bucket ID values, and then uses PIR to obtain only the bucket that their own template falls into. This shows a significant reduction in the individual query cost proportional to the number of buckets. Kulshrestha and Mayer [11] demonstrate this architecture concretely in the context of perceptual hash matching for end to end encrypted media using LSH bucketization to compress query space and computational PIR to retrieve the correct bucket privately. Servan-Schreiber et al. [12] for-

malized the communication efficiency gains of this pattern, showing that by partitioning LSH hash tables and amortizing PIR queries across partitions, the total server processing time can be held to linear time constraints.

LSH-Augmented PIR for Similarity Search

The key observation underlying our Bucketed PIR approach is that LSH maps similar high-dimensional vectors to the same bucket with high probability. Servan-Schreiber et al. [12] showed that issuing PIR queries over the resulting LSH hash tables achieves private approximate nearest-neighbor search with sublinear communication. SecureANNS [13] extended this framework to the semi-honest two-party setting, adapting LSH to select a small candidate subset. While these works focus broadly on nearest-neighbor discovery, our project explicitly adapts this combined primitive for the stricter requirements of biometric authentication. In our architecture, the client's goal is not merely to find the closest match in the database, but to privately retrieve their own specific enrolled template, decrypt it from the LWE-encrypted payload, and verify it against a tested 0.65 cosine similarity threshold. By tailoring LSH-augmented PIR to this binary accept/reject authentication model, we evaluate its viability for real-world identity verification rather than general similarity search.

Methodology

Dataset Selection

To evaluate the feasibility of biometric authentication with PIR, we constructed a mock database using the **Labeled Faces in the Wild (LFW)** dataset [14]. LFW is a widely used benchmark for facial recognition, containing over 13,000 images of more than 5,000 distinct people. For our testing, we used the 1,680 individuals in the dataset who had multiple images available. This allowed us to use one image as the "registered" template in the database while using the remaining images as "fresh face scans".

Feature Extraction and Quantization

We used a pretrained **FaceNet** model [15] to extract embeddings from each raw image. Our preprocessing pipeline involved identifying and cropping each image to frame the face, then using the model to convert each face into a 512-dimension vector of floating point objects where the distance between two vectors directly corresponded to facial similarity.

To reduce the PIR overhead caused by large database sizes, we optimized the size of our entries by quantizing each embedding vector. Each dimension was quantized to an 8-bit integer, resulting in a compact **512-byte representation** for every face in the database. This optimization significantly reduced the communication overhead during PIR retrieval while still preserving the signal to ensure accurate face matching.

Similarity Threshold

To determine whether two faces are a match, we decided to use the cosine similarity between the vectors. We used a set of independent images to verify the embedding and quantization pipeline and evaluated the cosine similarity between matching pairs (same person) and mismatched pairs (different people).

Based on the distribution of the cosine similarity scores, we arrived at a final threshold of **0.65** where a similarity exceeding the threshold indicates a match and a similarity lower than the threshold indicates different individuals.

Additionally, the LFW dataset contains many photos with varying degrees of image quality. To decrease the variance, we filtered the dataset to ensure a closer similarity scale between images. This allowed us to focus on testing our PIR scheme without worrying about guaranteed false rejections from overly blurry images or other quality impacting factors.

SimplePIR

SimplePIR [16] is a single-server computational PIR scheme based on the Learning with Errors (LWE) assumption. Following Kushilevitz and Ostrovsky's construction [4], with LWE parameters (n, q, χ) and plaintext modulus p, SimplePIR represents an N-element database as a (approximately) square matrix $\mathbf{D} \in \mathbb{Z}_p^{\sqrt{N}\times\sqrt{N}}$, which reduces retrievals to matrix-vector products and therefore allows for $O(\sqrt{N})$ communication. SimplePIR expands on it further by splitting the computation in two phases:

Offline. At setup, the server samples a shared public matrix $\mathbf{A} \in \mathbb{Z}_q^{\sqrt{N}\times n}$, and computes a $hint \in \mathbb{Z}_q^{\sqrt{N}\times n}$, $hint = \mathbf{D}\cdot\mathbf{A}$. Both $\mathbf{A}$ and the hint are sent to the client once and reused for all subsequent queries. This allows for most of the computation to be done upfront, amortizing and reducing per-query overhead.

Online. To retrieve entry i, the following process is followed:

1. The client decomposes i into row-col notation (r, c), and constructs a Regev-encrypted query

vector $qu \in \mathbb{Z}_q^{\sqrt{N}}$, $qu = \mathbf{A}s + e + \lfloor q/p \rfloor \cdot u_c$, where $s \leftarrow \mathbb{Z}_q^n$ is a fresh secret, e is a small noise vector, and u_c is one-hot encoding of the target column.

2. To answer this query, the server computes $ans \in \mathbb{Z}_q^{\sqrt{N}}$, $ans = \mathbf{D} \cdot qu$.

3. Finally, the client recovers the desired entry by using the hint and cancelling the LWE noise: $\mathbf{D}[r][c] = \text{round}_\Delta(ans[r] - hint[r] \cdot s)/\Delta$, where $\Delta = \lfloor q/p \rfloor$

Direct PIR The 'direct' PIR method for biometric authentication employs a fairly straightforward flow (Fig.1): suppose a user claims the identity of a person i (an index), using SimplePIR, the client queries the server for the known registered biometric embedding that corresponds to user i; once received, the client computes the cosine similarity score and, given a predetermined threshold, determines whether to authenticate the user or not.

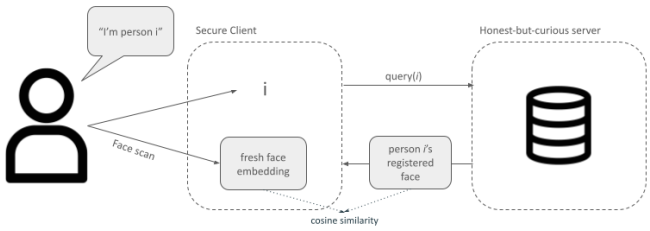


Figure 1: Overview: Authentication process with direct PIR

Database Matrix For a person with index i, one corresponding 512-dimensional FaceNet vector (quantized to int8 values) is packed into $512 \cdot 8/64 = 64$ `64-bit` values and included into a flat database, such that person i's face embedding belongs in the indexes $[64i,\ 64(i+1)-1]$. This flat array of $64N$ `uint64` values is passed to SimplePIR's `MakeDB` with entry width $d = 64$ bits, which reshapes it internally into an approximately square matrix.

Retrieval Given that a face is split into 64 entries in the database, retrieving a full embedding results in 64 SimplePIR queries to the server.

Parameters For this project, the scheme is instantiated with LWE security parameter $n = 1024$ and ciphertext modulus $\log q = 32$, giving plaintext modulus $p = 991$ and noise parameter $\sigma = 6.4$.

Bucketed PIR In comparison, to overcome the known computational and bandwidth limitations of querying the entire biometric database, we also implement a Bucketed Private Retrieval architecture. This approach integrates the previously defined LSH index with the SimplePIR protocol to securely and efficiently retrieve localized clusters of candidate identities.

By routing queries to localized clusters, the server avoids computing matrix multiplications over the entire dataset. However, our empirical evaluation at the $N = 1,497$ scale reveals that the constant-factor overheads of this approach currently outweigh its theoretical benefits for small datasets.

Cryptographic Parameter Alignment A subtle challenge in this implementation concerned the plaintext mod constraints. The LWE lattice parameters underlying SimplePIR enforce a strict plaintext modulus of $p = 991$. The typical PIR optimization techniques of packing multiple 8-integers into a single 64-bit word resulted in data corruption, which we overcame by adopting a one-to-one mapping, assigning each unsigned 8-bit facial feature to its own 32-bit lattice word. While this increased storage overhead, it was the only approach that guaranteed absolute mathematical fidelity throughout the pipeline.

Database Matrix Allocation The LSH-partitioned buckets are flattened sequentially and ingested by the SimplePIR native setup routines. The cryptographic engine determines optimal matrix dimensions autonomously.

Blind Retrieval The online retrieval phase is executed as a strict two-party blind protocol: query formulation and biometric verification.

Query Formulation: Using the shared LSH hyperplanes, the client first determines which bucket ID is most likely to contain the target identity. The client then encodes this bucket index into a homomorphic query vector using the shared lattice parameters.

Decryption and Biometric Verification: Upon receiving the payload, the client decrypts it using the local secret key and server-provided hint, recovering the exact sequence of unsigned 8-bit integers. The client then reshapes this flat array into 512-dimensional FaceNet vectors and shifts them back to the signed $[-128, 127]$ integer range. Finally, cosine similarity is computed between the query vector and the retrieved candidates. Any value below a predefined threshold, 0.65 in this project, gets rejected, and all others are accepted as positive matches, completing client authentication.

Results

Direct PIR

We evaluated Direct PIR across five database sizes: 1,497 people derived from the Labeled Faces in the Wild (LFW) [14] dataset, and four augmented databases of 5,000, 10,000, 50,000, and 100,000

people. Augmented entries use randomly generated `int8` embeddings and are never queried for verification; they exist only to assess Direct PIR's scaling behavior. Figure 2 shows four metrics across database sizes.

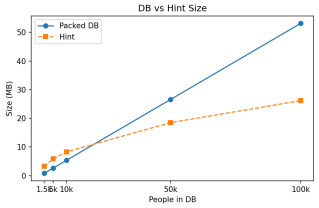
Server Setup Time (Fig. 2a) Setup time grows approximately linearly with database size, reaching 8.2 seconds to build a database of 100k people; this trend is consistent with SimplePIR's theoretical $O(N)$ cost of $2nN$ operations for an N-entry database. For databases with frequent updates, linear setup cost is a practical limitation.

Database vs. Hint Size (Fig. 2b) As the database grows linearly, hint size grows as $O(\sqrt{N})$, reaching 26.2MB for the 100k-people database. The hint actually exceeds the database at small scale due to constraints imposed by LWE parameters; the crossover occurs at $\sim$20k people.

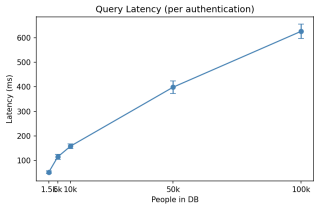
Query Latency (Fig. 2c) Latency was measured for the full authentication attempt, i.e. 64 SimplePIR `query`+`answer`+`recover`. It ranged from 52ms at 1,497 people to 626ms. While the server answer step for a N-entry database requires $2N$ operations (i.e. linear in N), the empirical curve likely looks sublinear due to the aggregation with client-side operations, and other fixed overhead.

Online Communication Cost (Fig. 2d) Similarly to latency, this was measured for a roundtrip authentication attempt. Growth follows $O(\sqrt{N})$, consistent with SimplePIR's theoretical.

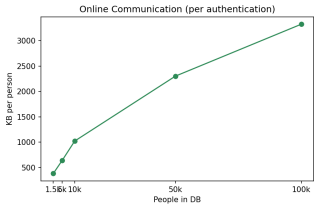
(a) Server setup time (seconds)



(b) Database vs. hint size (MB)



(c) Query latency per authentication



(d) Online communication cost per authentication

Figure 2: Direct PIR metrics

Comparison to Insecure Baseline. To determine the cost of the privacy guarantee, we compare Direct PIR with an insecure direct lookup that returns `DB[i]` without SimplePIR. Given that this insecure implementation would not require set up, we can only compare query latency and online communication costs. For all five databases, average latency was always < 0.001 milliseconds, and online communication around 4KB (one index plus the 512 `int8` face embedding).

Bucketed PIR

The Bucketed Locality Sensitive Hashing PIR architecture was designed to achieve computational scalability by reducing the homomorphic search space from $O(N)$ to $O(N/B)$. The architecture was evaluated across its offline construction efficiency, bandwidth footprint, and online retrieval latency.

During the offline phase, the server takes in the multi-template face dataset and executes the LSH partitioning and matrix generation, averaging 983ms across 2814 samples. The packed matrix size reached 9.9MB on average, which to note is larger than the Direct PIR baseline. This inflation is a direct consequence of our conservative parameter alignment described above.

In the online phase, end-to-end client queries averaged 29 seconds. This protocol successfully recovered uncorrupted payloads and executed robust threshold matching, proving the functional correctness of the architecture. However, the 29-second latency makes it impractical for real-time authentication in its current state.

We attribute this latency primarily to the client-side decryption and reshaping pipeline. Because the data was not densely packed, the client must execute a disproportionately high number of lattice decryption operations to recover the full 512-dimensional vector. While Direct PIR significantly outperformed Bucketed PIR at this small scale (52ms vs. 29s), Bucketed PIR's server-side computation scales at $O(N/B)$. In our motivating scenario of national-scale databases, we believe the computational cost of Direct PIR would eventually exceed the fixed decryption overhead of the Bucketed architecture.

Dataset Variance and Real-World Applicability The online retrieval phase exposed a vulnerability tied to the unconstrained nature of the LFW dataset. Although we filtered the dataset to a 0.65 simiarilty scale, because the remaining images still had differences in lighting, pose, or quality, the user's fresh query vector was still able to shift across a LSH hyperplanes. When this happens, the query maps to an incorrect bucket and is rejected before cosine similarity is applied, raising the false rejection rate.

However, this limitation does not invalidate the architecture for our motivating scenario of remote

authentication at airport security kiosks. Unlike the unconstrained LFW dataset, airport kiosks represent highly controlled, cooperative biometric environments. Users are required to look at a fixed-angle camera, with bright frontal illumination. In such an environment, the biometric capture variance drops, meaning a user's query vector will reliably and consistently map to the correct LSH bucket, keeping the false rejection rate within acceptable operational bounds.

Conclusion

This paper evaluated SimplePIR as a practical primitive for privacy-preserving biometric authentication through two implementations, Direct PIR and Bucketed PIR, over FaceNet embeddings from the Labeled Faces in the Wild dataset.

Direct PIR provides a strong and well-characterized privacy guarantee at a quantifiable cost. Authentication latency ranged from 52ms at 1,497 people to 626ms at 100,000 people, where online communication reached $\sim$3.3MB, and server setup took 8.19s. Communication metrics followed $O(\sqrt{N})$.

Bucketed PIR aimed to reduce the per-query server search from $O(N)$ to $O(N/B)$ via LSH. Evaluated on the 1,497-person database, it averaged 982ms for offline setup, and 29 seconds per online query; significantly higher than Direct PIR's 121ms setup time and 52ms query latency on the same database. The additional overhead likely stemmed from LSH partitioning, and payload decryption. The high variance in the LFW dataset caused embeddings to shift across LSH hyperplanes, raising false rejection rates; this limitation is dataset-specific and would be mitigated in controlled environments, such as airport kiosks.

Both approaches fall short of deployment readiness in their current forms. Direct PIR's linear setup cost makes frequent database updates expensive, and Bucketed PIR's 29-second query latency is impractical for interactive use. Direct PIR is viable for medium-scale deployments with infrequent database updates. Bucketed PIR's architecture is better suited for the 1:N identification setting where a client cannot assume a known index, provided biometric capture conditions are controlled enough to improve LSH bucket stability.

Future Work

Although we demonstrate the feasibility of integrating PIR into biometric authentication systems, there are still several paths to explore surrounding optimization and scalability. One of the primary limitations of our current Bucketed PIR implementation is the **server initialization overhead**. During its evaluation, the server was being rebuilt for each query, causing significant latency. In the real world, the server would not be rebuilt during each query, so future iterations should use a persistent server process to avoid redundant overhead and more accurately represent performance.

Additionally, our evaluation was limited by the number of images in the LFW dataset. Although we scaled our mock database up to 100,000 entries using randomly generated embeddings that were never queried, our tests were limited to the 1,680 people (pre-filtered count) who had multiple photos in the dataset. To accurately assess the scalability of these approaches, future research should test them on **massive datasets** with millions of faces to better simulate real-world applications where this protocol is most useful. Testing on larger datasets will better demonstrate the tradeoffs our approaches encounter and reveal how quickly the PIR overhead grows.

Finally, we would like to benchmark our results with other **state-of-the-art PIR implementations** such as SpiralPIR, XPIR, and Piano. Although our Direct PIR implementation using SimplePIR was successful and performed well with our limited test data, we are curious how other protocols may differ and what advantages each provides. Protocols like **SpiralPIR** [17] use Fully Homomorphic Encryption to reduce query size and minimize communication overhead, while protocols like **Piano** [18] achieve sublinear sever computation only relying on one-way functions. Exploring and comparing these protocols will help us determine the most efficient option for biometric authentication use cases.

Acknowledgments

We would like to express our sincere gratitude to Professor Devadas for his insightful instruction this semester, and for his invaluable insight into various methodological approaches that we incorporated into this project.

We are especially thankful to our head TA Kevin for his exceptional guidance and dedication towards our project, asking us guiding questions every time we met to help us more deeply understand the workabouts inside the project. We would also like to thank Simon for listening in and asking clarifying questions which shaped the way we presented our work.

Finally, we extend our appreciation to the entire course staff of 6.5610 for their consistent support and assistance throughout both the class and this project.

Contributions

This project was done in collaborations by Zhiron Wu, Robin Xiong, and Catherine Zhu. Although each section was communicated clearly to each other throughout the timeline of the project, Catherine focused on dataset preprocessing, Zhiron worked on the pipeline for direct PIR, and Robin worked on the pipeline for bucketed PIR.

In the paper, Catherine wrote the Abstract, Introduction, Methodology: Dataset Selection, and Future Work sections. Zhiron wrote Methodology: SimplePIR, Methodology: Direct PIR, Results: Direct PIR, and Conclusion. Robin wrote Related Work, Methodology: Bucketed PIR, Results: Bucketed PIR, and Results: Dataset Variance and Real-World Applicability. The entire paper was reviewed for grammatical issues and general language smoothing before submission by all members.

References

[1] K. Prakasha and S. Udaya, "Privacy-preserving techniques in biometric systems: Approaches and challenges," *IEEE Access*, vol. PP, pp. 1–1, 01 2025.

[2] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, pp. 4 – 20, 02 2004.

[3] B. Chor, O. Goldreich, and E. Kushilevitz, "Private information retrieval," vol. 45, pp. 41–50, 11 1995.

[4] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval," in *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, FOCS '97, (USA), p. 364, IEEE Computer Society, 1997.

[5] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-Preserving Face Recognition," in *Privacy Enhancing Technologies* (I. Goldberg and M. J. Atallah, eds.), (Berlin, Heidelberg), pp. 235–253, Springer, 2009.

[6] F. Kerschbaum, M. Atallah, D. M'Raïhi, and J. Rice, "Private Fingerprint Verification without Local Storage," vol. 3072, pp. 387–394, Jan. 2004.

[7] S. F. Shahandashti, R. Safavi-Naini, and P. Ogunbona, "Private Fingerprint Matching," 2012. Publication info: Published elsewhere. ACISP 2012.

[8] J. Bringer, H. Chabanne, D. Pointcheval, and Q. Tang, "Extended Private Information Retrieval and Its Application in Biometrics Authentications," in *Cryptology and Network Security* (F. Bao, S. Ling, T. Okamoto, H. Wang, and C. Xing, eds.), (Berlin, Heidelberg), pp. 175–193, Springer Berlin Heidelberg, 2007.

[9] J. Bringer and H. Chabanne, "An Authentication Protocol with Encrypted Biometric Data," in *Progress in Cryptology – AFRICACRYPT 2008* (S. Vaudenay, ed.), (Berlin, Heidelberg), pp. 109–124, Springer, 2008.

[10] R. Ostrovsky and W. E. Skeith, "A Survey of Single-Database Private Information Retrieval: Techniques and Applications," in *Public Key Cryptography – PKC 2007* (T. Okamoto and X. Wang, eds.), (Berlin, Heidelberg), pp. 393–411, Springer, 2007.

[11] A. Kulshrestha and J. Mayer, "Identifying Harmful Media in End-to-End Encrypted Communication: Efficient Private Membership Computation," pp. 893–910, 2021.

[12] S. Servan-Schreiber, S. Beyzerov, E. Yablon, and H. Park, "Private Access Control for Function Secret Sharing," 2022. Publication info: Published elsewhere. Minor revision. IEEE Symposium on Security and Privacy 2023.

[13] S. Song, L. Liu, R. Chen, W. Peng, and Y. Wang, "Secure Approximate Nearest Neighbor Search with Locality-Sensitive Hashing," in *Computer Security – ESORICS 2023: 28th European Symposium on Research in Computer Security, The Hague, The Netherlands, September 25–29, 2023, Proceedings, Part III*, (Berlin, Heidelberg), pp. 411–430, Springer-Verlag, Sept. 2023.

[14] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," in *Workshop on Faces in 'Real-Life' Images:*

Detection, Alignment, and Recognition, (Marseille, France), Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Oct. 2008.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 815–823, IEEE, 2015.

[16] A. Henzinger, M. M. Hong, H. Corrigan-Gibbs, S. Meiklejohn, and V. Vaikuntanathan, "One server for the price of two: Simple and fast single-server private information retrieval," in *32nd USENIX Security Symposium (USENIX Security 23)*, (Anaheim, CA), USENIX Association, Aug. 2023.

[17] S. J. Menon and D. J. Wu, "Spiral: Fast, high-rate single-server PIR via FHE composition." Cryptology ePrint Archive, Paper 2022/368, 2022.

[18] M. Zhou, A. Park, E. Shi, and W. Zheng, "Piano: Extremely simple, single-server PIR with sublinear server computation." Cryptology ePrint Archive, Paper 2023/452, 2023.