Privacy Preserving Surveillance from Fuzzy Labeled Private Set Intersection

Erkam Uzun, Simon P. Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee
Georgia Institute of Technology

ABSTRACT

The explosive growth of biometrics use (e.g., in surveillance) poses a persistent challenge to keep biometric data private without sacrificing the apps' functionality.

We consider private querying of a real-life biometric scan (e.g., a person's face) against a private biometric database. The querier learns only the label(s) of a matching scan(s) (e.g. a person's name), and the database server learns nothing.

We introduce *Fuzzy* Labeled Private Set Intersection (FLPSI), a primitive computing the intersection of noisy input sets by considering closeness/similarity instead of equality.

Our FLPSI protocol's communication is *sublinear* in database size and is concretely efficient. We implement it and apply it to facial search by integrating with our fine-tuned toolchain that maps face images into Hamming space. FLPSI achieves high performance with concretely small network usage: for a 1M-row database, online time is 1.66s (WAN) and 1.46s (fast LAN) with 40.8MB of data transfer in online phase and 37.5s in offline precomputation. This improves the state-of-the-art work (SANNS) by 9 − 25× (on WAN) and 1.2 − 4× (on fast LAN).

CCS CONCEPTS

• **Security and privacy** → **Cryptography**; **Management and querying of encrypted data**.

KEYWORDS

privacy preserving, surveillance, homomorphic encryption

ACM Reference Format:
Erkam Uzun, Simon P. Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee, Georgia Institute of Technology . 2021. Privacy Preserving Surveillance from Fuzzy Labeled Private Set Intersection. In *PPML '21: Privacy Preserving Machine Learning ACM CCS 2021 Workshop, November 19, 2021, Virtual.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recent advances in deep learning (DL)-based biometric identification have made possible real-time identification of persons in footage collected by surveillance equipment. The trend toward real-time surveillance in public and private places (e.g., streets, city halls, airports, retail stores, pharmacies, gas stations etc.) has immense benefits for public safety or customer convenience. However, adoption of these technologies come at a significant privacy cost, which raises serious objections.

To our knowledge, existing biometrics surveillance systems have the following major challenges. First, vendors store and process the collected biometric data on their server in plaintext for the ease of deployment and practicality. Second, people cannot opt-out of these systems, since video footage (or any captured faces) are directly uploaded to a remote server.

Identifying "persons of interest" may be warranted [31], but tracking *everybody else* in the process may open the doors to illegitimate surveillance and certain human right abuses [33]. In response, privacy stakeholders are pressing for a moratorium on permanent adoption of these systems, and in fact they have already succeeded in banning facial surveillance in several countries and U.S. states [4, 30, 32].

In this paper, we propose a middle ground solution, *privacy-preserving biometric search.* Here the server $\mathcal{S}$ holding a large biometric database with corresponding labels (e.g., identities) should learn nothing about the query or the result, while the querier (client C) should learn nothing about the database besides the label(s) of the query's match(es).

A similar problem of exact private match is extensively studied in a variety of scenarios (e.g., contact list discovery and online dating services), and can be achieved via (labeled) private set intersection (LPSI), a standard primitive [9, 10, 20, 22]. Even though the state-of-the-art CHLR18 [9] achieves a practical efficiency with communication costs sublinear to DB size, LPSI cannot directly be applied to our problem because it targets *exact* matches, while biometrics are noisy (e.g., due to lighting, imprecise scans, etc.).

We introduce FLPSI: a *fuzzy LPSI* protocol for fast privacy-preserving biometric search[1]. We address a number of technical challenges in protocol/definition design and formal proofs.

To our knowledge, none of the prior work related to fuzzy matching achieves practical efficiency for real-time surveillance, mainly because of communication growing (at least) linearly with database size. For example, two protocols of the state-of-the-art (SANNS [8]) require 1.7-5.4 GB communication and 15.1-41.7 sec. online response times over WAN per query over a million-row database.

We follow a much more scalable approach that reduces our fuzzy matching problem to an easier exact-matching subproblems that could be solved with communication cost sublinear in DB size, by leveraging optimizations of the state-of-the-art (L)PSI techniques [9, 10]. The crux of our solution is twofold. First, we translate the closeness (e.g., in Euclidean space) of two biometrics into a *t-out-of-T* set-based matching without sacrificing accuracy. That is, we encode a given biometric input into a set of T items, s.t. the two sets

[1] Please refer the full version of this paper [34] for more details.

PPML '21, November 19, 2021, Virtual

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

will likely have at least t *exactly* common items iff the biometrics are of the same person. Second, we build an efficient threshold set-matching protocol from fully homomorphic encryption (FHE), garbled circuits (GC) and *t-out-of-T* secret sharing, and solve several challenges in definitional approach.

1.1 Summary of Our Contributions

- We build a FLPSI protocol using the AES blockcipher, homomorphic encryption, garbled circuits and *t-out-of-T* secret sharing. We prove the security in the semi-honest model.
- We show how to interpret closeness (e.g., in Euclidean space) between biometric inputs as *t-out-of-T* exact set-item matchings without sacrificing the accuracy.
- We give simulation-based FLPSI security definition (prior definitions of fuzzy primitives are game-based).
- We introduce a number of optimizations, in addition to the prior (L)PSI techniques we use.
- We achieve 1.66s online running time over WAN with 40.8MB transfer per query over a million-row database.
- We systematically compare our design with prior art, and outperform all of them in their best settings, often by several orders of magnitude both in run time and communication. For example, on the largest dataset (of 10M records), we speed up by a factor of 3-33× and decrease the overall data communication by a factor of up to 48-452× compared to the two protocols of the state-of-the-art, SANNS [8].

2 OVERVIEW AND TECHNICAL DETAILS

Here we review existing non-private (plaintext) fuzzy matching algorithms and building privacy protection into them. We define, construct and prove the security of FLPSI in the extended version.

2.1 Plaintext Fuzzy Matching

Existing facial surveillance systems, informally, work as follows. A client C captures facial images of people from a surveillance video footage, then transmits the biometric data to a server $\mathcal{S}$ with transport encryption, and $\mathcal{S}$ receives the data in plaintext. Then, the server uses a DL system to turn raw biometric readings into embedding vectors with a (probabilistic) guarantee that two such vectors will be close in Euclidean distance iff they are from the same person. If the server finds such a close biometric entry in its database, it returns the label (e.g, identity) of the entry to the client. Otherwise, it returns "*no match*" result to the client.

Privacy concerns. Since the data is processed in plaintext by the server, it achieves maximal privacy, while the client achieves none. Next, we discuss achieving maximal client privacy as well.

2.2 Private Fuzzy Matching

Our goal is to build a protocol that reveals labels of query matches only to C, while maintaining confidentiality of C's query and $\mathcal{S}$'s database. To achieve this, C and $\mathcal{S}$ can locally compute DL embeddings from their biometric data, then apply standard MPC tools to compute Euclidean (or cosine similarity) distance between the C's query and each of the $\mathcal{S}$'s database items [2, 3, 13, 15, 24]. However, this does not scale. Our much more scalable approach is based on a *t-out-of-T matching* scheme, described in detail next. Fig. 1 shows a high-level overview of our FLPSI protocol.

Binary encoding. To accommodate *t-out-of-T* matching, we first address the incompatibility between DL embeddings (operating in Euclidean space) and the crypto components (operating in Hamming space) of our protocol. (Operating in Hamming space, e.g., computing closeness is *much* cheaper in MPC). To do this, parties additionally apply a space mapping function, which is based on locality-sensitive hashes [6, 17, 19, 23], to convert the embedding vectors into bio-bit vectors (x_i and y) with the desired property (they are Hamming-close if they originate from the biometrics of the same person). For lack of space, we refer [35] for more details about our space mapping technique. We will refer to the set of functions converting biometric data into bio-bit vectors, as "$Encode(.)$".

C and $\mathcal{S}$ proceed as follows after encoding their biometric data into bio-bit vectors y (held by C) and $x_i \in \mathcal{X}$ (held by $\mathcal{S}$).

- **Subsampling:** generate T random subsamples of y and each x_i bio-bit vectors (in the same way, e.g., $x_{21} = x_2 \wedge mask_1$), s.t. at least t of them will be the same iff y and a x_i belong to the same person (if bio-bit vectors are Hamming-close, this can be achieved whp);
- **Secret sharing:** generate *t-out-of-T* secret shares of the label l_i (e.g., identity) of each $x_i \in \mathcal{X}$ (each share is associated with a subsample of x_i), s.t. any t shares can reconstruct l_i; Note that $\mathcal{S}$ attaches an agreed-upon token 0^λ, where λ is a security parameter, to each label l_i before secret sharing it. Then, C can verify if any set of t shares (obtained via a single STLPSI execution) correctly reconstruct a valid label.
- **STLPSI:** interactively execute a private *t-out-of-T* matching protocol (*Set Threshold LPSI, or STLPSI*) on the C's subsample set and the $\mathcal{S}$'s sets of (subsample, secret share) pairs[2]: the label l_i of an $x_i \in \mathcal{X}$ is revealed to C iff at least t of the subsamples of y and x_i are equal (which means C obtains shares of matching subsamples of x_i).

2.2.1 *Our Solutions to Technical Challenges.*

Now we discuss the most interesting technical challenges.

Subsample confidentiality As described above, C learns the subsamples (and respective subsampling masks), which may help C learn something additional about database. For example, in case of a false-positive match, the semi-honest C will now learn with confidence positions in bio-bit vector, thereby learning the original biometric, which may not be included in the result set. Further, it may be the case (and publicly known) that faces in $\mathcal{S}$'s database are similar (e.g. manifested by certain positions of the bio-bit vectors being equal). A match from C's query will inform the malicious C how to set the bits of his next query so as to improve his chance of "hitting" a face in database (a false match).

We can resolve this by operating over encrypted subsamples only. For this, $\mathcal{S}$ chooses the random subsampling/projection masks and an AES encryption key $k_{\mathcal{S}}$. Then $\mathcal{S}$ via MPC allows C to compute the AES-encryptions of masked projections on the C's query bio-bit vector y, while keeping the projection masks and $k_{\mathcal{S}}$ secret from C. The server efficiently computes AES-encryptions of masked projections on its large database non-interactively in $O(|\mathcal{X}|)$. Note that $\mathcal{S}$ has to refresh these masks and keys for each execution.

[2] Note that the secret shares are now treated as *labels* in the STLPSI.

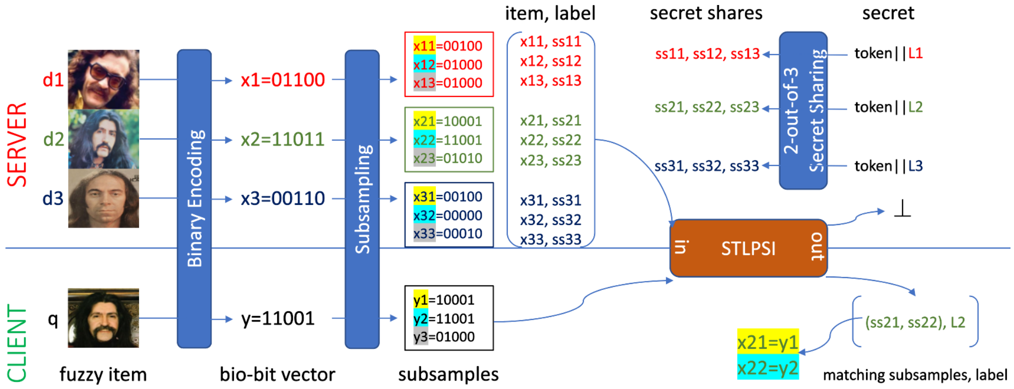

Figure 1: Overview of FLPSI. For clarity, subsampling is depicted without AES encryption and 2PC.

Concealing partial matches in single execution. (L)PSI protocols (e.g., [9, 10, 20, 22]) do not directly implement the above STLPSI functionality since they, by design, reveal partial (below-threshold t) matches. We resolve this by building efficient STLPSI from *t-out-of-T* secret sharing and FHE, based on prior (L)PSI works (e.g., CHLR18 [9]).

Concealing partial matches in repeated executions. This subtle issue arises when generated shares are not refreshed between queries, and C may collect threshold t shares *across* queries. We resolve this by carefully resetting secret shares, subsampling masks and keys in each execution.

Novel definitional approach. In MPC, the preferred simulation-based security definitions offer clean and composable guarantees. At the same time, they require precise specification of ideal-world behavior, which we (as a community) do not know how to achieve for biometric functions. Because of this, biometric authentication definitions are usually game-based and not composable, but which allow to *bound*, rather than *precisely specify* adversary success.

One of our contributions is a *novel definitional approach*, which allows the best of both worlds: our definition is indeed simulation-based, and yet we bound adversary success rather than exactly specifying it. We reconcile the yin and yang and achieve the best of both by defining the ideal FLPSI functionality via a reference to a real FLPSI protocol. Namely, we will say that ideal functionality outputs whatever the real protocol formally outputs. While at the first glance this may seem a tautology, this approach does provide a guarantee that nothing beyond the explicit protocol output is revealed. Now we are in a good place, since we can easily control explicit protocol output by specifying the *correctness* property.

We believe this definitional approach can serve as a template in defining primitives in the biometric space.

3 EVALUATION

In this section, we introduce our environmental setup and databases, and then systematically compare FLPSI to the prior art.

3.1 Environment and Implementation Details

We use an Azure F72s_v2 instance, which has 72 virtual cores equivalent to that of 2.7 GHz Intel Xeon Platinum 8168 CPU and 144 GB of RAM each. We also have two sets of experiments: for *fast* and *slow* network connections between C and $\mathcal{S}$. While the former has 500 MB/s connection with 0.5 ms latency, the latter is having 40 MB/s with 34 ms latency. We use Ubuntu 18.04 in this instance. *Note that, even though, our design does not require a fast network connection or high number of threads, we use above environment for creating an identical comparison setting with the state-of-the-art* [8].

Protocol	Communication	Computation
FLPSI	$O(\frac{NT}{mB}\ell)\approx O(T\ell)$	$O(\frac{NT}{m})$
CEC [5]	$O(N\lvert\mathbb{F}_{\mathcal{P}}\rvert\ell)$	$O(N(\lvert\mathbb{F}_{\mathcal{P}}\rvert+T)T'_{\epsilon})$
YSPW [38]	$O(NT^2\ell)$	$O(N(poly(T)+T^2T'_{\epsilon}))$
CH_1 [11]	$O(NT\ell)$	$O(N(\binom{T}{t}poly(T)+TT'_{\epsilon}))$

Figure 2: Comparing FLPSI with existing *t-out-of-T* protocols. Only the dominant terms are kept. ℓ is the size of a ciphertext in the chosen encryption scheme. T'_{ϵ} is the time needed for all homomorphic operations in a single cycle.

We implement our protocol on top of the FHE library SEAL v3.5 [26], through Brakerski/Fan-Vercauteren (BFV) scheme [14]; Yao's Garbled Circuits (GC) using the EMP toolkit [36]; and Shamir's secret sharing [27]. To extract embedding vectors from facial images, we use the Python implementation of FaceNet[3] (with the Inception-Resnet-V1 architecture [29]) after aligning faces, as recommended in [39]. We parameterized our system to achieve the same false-match and false-non-match rates with the compared works while achieving at least a 128-bit security level on the FHE-based construction as recommended in [7]. We used a 2-out-of-64 matching scheme in our STLPSI and secret sharing constructions.

3.2 Datasets

For our comparative analysis, we use AT&T [25] and Deep1B [1] datasets, which are used in prior art. Note that we use these datasets in the same way as they are used in the prior art. AT&T[4] includes 400 facial images from 40 people, where 8 faces of each (320 in total) are kept as database items and 2 faces of each are queried. Deep1B contains a billion image descriptors (each 96 dimension vector), which is generated by passing images through a deep neural network [1]. We use the original query set, which includes 10 thousand data points, published by the authors[5]. And, we conduct queries over two subsets of Deep1B that consist of randomly selected one million and 10 million entries (labeled as Deep1B-1M

[3] https://github.com/davidsandberg/facenet
[4] https://www.kaggle.com/kasikrit/att-database-of-faces
[5] http://sites.skoltech.ru/compvision/noimi/

Protocol	Deep1B-1M				Deep1B-10M			
	Communication		Response time (fast/slow)		Communication		Response time (fast/slow)	
	Total	Saving	(seconds)	Speed up	Total	Saving	(seconds)	Speed up
FLPSI	40.8 MB	-	1.46/1.66	-	128 MB	-	12.7/13.5	-
SANNS-linear	5.39 GB	132×	5.79/41.7	3.97/25.1×	57.7 GB	452×	73.1/446	5.76/33.0×
SANNS-approx	1.72 GB	42×	1.70/15.1	1.16/9.09×	6.07 GB	48×	5.27/41.8	0.41/3.10×

Figure 3: Comparing FLPSI to two protocols of SANNS [8]. Best achieved response times are reported for fast/slow networks.

and Deep1B-10M, respectively). We treat Deep1B descriptors as embedding vectors in our pipeline since it is not a facial dataset.

3.3 End-to-end Comparison with Prior Art

In this section, we systematically compare FLPSI with previous *private fuzzy matching* protocols. Considering their functionality and security guarantees for our application scenario, we group prior art in two categories: i) *threshold matching* and ii) *k-nearest neighbor search.* In (i), as in our work, $\mathcal{S}$ may return empty result (depending on the false-match error) to C if no close entry exists in the database. In (ii), $\mathcal{S}$ always guarantees to return k database entries to C regardless of the query. While (ii) is a different functionality, we compare our work with protocols in both categories, as the state-of-the-art (SANNS [8]) in (ii) is also faster than protocols in (i), and is the fastest among protocols "close enough in spirit".

3.3.1 Comparison to Threshold Matching Approaches. Prior art either a) applies thresholding to computed Euclidean (or Hamming and cosine similarity) distance, or b) runs *t-out-of-T* matching between query and database (feature) vectors. Though they satisfy the functionality requirement and security guarantees for our application, none of them propose a practically applicable system for a real-time surveillance task.

Distance thresholding approaches. We compared concrete costs of FLPSI to prior work [2, 3, 13, 15, 21, 24, 37]. Note that the cited works report communication and computation costs linear in the database size. They achieve between 1.7-99.2 sec. response times and 2.8-35.2 MB network overheads per query over AT&T database. We achieve **121-7086×** faster response time (14 ms. per query) and **7.18-90.3×**less communication for the *same* database.

t-out-of-T matching approaches. Systems [5, 11, 38] (referred as CH$_1$[6], YSPW, CEC, resp.) are existing, secure, *t-out-of-T* protocols. Fig. 2 compares asymptotic communication and computation complexity of [5, 11, 38] to our system. FLPSI behaves better both in computation and communication than CH$_1$, YSPW, and CEC protocols, as both of their communication and computation complexities are linear in database size.

3.3.2 Comparison to kNNS Approaches. We emphasize that "k-nearest neighbor search" protocols solve a somewhat related, yet different problem, and do not meet the security guarantees we consider. Nevertheless, we compare them to FLPSI because we wish to present a broader perspective and to illustrate that our work is more efficient not only than protocols for our exact problem, but than any prior work "close enough in spirit."

We compare our design with Chen et al. [8]'s two protocols since, to our knowledge, they are the fastest protocols compared to all other kNNS approaches [12, 16, 18, 28], which do not use a trusted third-party in their pipelines. SANNS propose an optimized linear scan (SANNS-linear) and an approximate search (SANNS-approx) protocols, which are built upon additive homomorphic encryption, garbled circuits and oblivious read only memory, to conduct secure kNNS over large databases. To conduct an almost identical comparison, we evaluate FLPSI on the same Azure instances with the same *fast/slow* network connections, as introduced in Sect. 3.1, and over the same image datasets: Deep1B-1M and Deep1B-10M.

Communication and computation costs. Fig. 3 compares total communication overheads and the best achieved response times through the fast/slow networks for the both database sizes. Due to our sublinear communication, FLPSI decreases required bandwidth by **132-452×** and **42-48×** (depending on the database size) compared to SANNS's linear and approximate protocols, respectively. This implies significant improvement in wall-clock time, especially on slower networks. In fact, SANNS outperforms FLPSI only on Deep1B-10M dataset, with *fast* network connection, and via its approximate algorithm. For instance, the best response time of SANNS-approx protocol increases from 1.7 to 15.1 sec. as we switch the network from fast to slow connection. Similarly, SANNS-linear's performance decreases even more in the same situation, as it has more data overhead than their approximate protocol. On the other hand, FLPSI preserves its performance regardless of the network connection, as it has 128 MB of communication overhead even for a database of 10 million entries. Overall, we achieve up to **5.8/33×** and **1.2/9.1×** faster response times compared to SANNS's linear and approximate protocols, respectively, on the fast/slow networks.

4 CONCLUSIONS

We introduce FLPSI, fuzzy labeled private set intersection, and propose an efficient construction. In FLPSI, client C holds a biometric query and server $\mathcal{S}$ holds a labeled biometric database, where labels may be, e.g., persons' identities. In FLPSI, C learns the label *iff* the query is in the database, and $\mathcal{S}$ will learn nothing. Our definitional approach uniquely combines the properties of game-based and simulation-based definitions, and can be useful in other settings.

Designing an efficient protocol for FLPSI is challenging mainly due to the need to manage the noisiness of biometric data. We realize FLPSI in the semi-honest model from a blockcipher, garbled circuits, secret sharing, and fully homomorphic encryption.

FLPSI achieves *sublinear* communication cost relative to the database. Our experiments show that our solution scales well to massive datasets including up to 10 million entries. Additionally, our comparative results show that i) FLPSI achieves up to 48-452× less communication cost and ii) up to 3.1/33× faster response times compared to protocols from the state-of-the-art on a database of 10 million entries. Notably, FLPSI has a major advantage over prior art by not relying on high speed network connection for efficiency.

[6] Ye et al. [38] break the security of the second protocol from [11].

REFERENCES

[1] Artem Babenko and Victor Lempitsky. 2016. Efficient indexing of billion-scale datasets of deep descriptors. In *IEEE CVPR*.

[2] Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Fabio Scotti, et al. 2010. Privacy-preserving fingercode authentication. In *MM&Sec*.

[3] Marina Blanton and Paolo Gasti. 2011. Secure and efficient protocols for iris and fingerprint identification. In *ESORICS*. Springer.

[4] Business Insider. [n.d.]. https://www.businessinsider.com/senate-bill-sanders-merkley-ban-corporate-facial-recognition-without-consent-2020-8.

[5] Ioan Calapodescu, Saghar Estehghari, and Johan Clier. 2017. Compact fuzzy private matching using a fully-homomorphic encryption scheme. US Patent 9,749,128.

[6] Moses S Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *STOC*.

[7] Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Jeffrey Hoffstein, Kristin Lauter, Satya Lokam, Dustin Moody, Travis Morrison, et al. 2017. Security of homomorphic encryption. *HomomorphicEncryption.org, Tech. Rep* (2017).

[8] Hao Chen, Ilaria Chillotti, Yihe Dong, Oxana Poburinnaya, Ilya Razenshteyn, and M. Sadegh Riazi. 2020. SANNS: Scaling Up Secure Approximate k-Nearest Neighbors Search. In *USENIX Security*.

[9] Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal. 2018. Labeled PSI from Fully Homomorphic Encryption with Malicious Security. In *CCS*.

[10] Hao Chen, Kim Laine, and Peter Rindal. 2017. Fast private set intersection from homomorphic encryption. In *CCS*.

[11] Lukasz Chmielewski and Jaap-Henk Hoepman. 2008. Fuzzy private matching. In *ARES*.

[12] Daniel Demmler, Thomas Schneider, and Michael Zohner. 2015. ABY-A framework for efficient mixed-protocol secure two-party computation.. In *NDSS*.

[13] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. 2009. Privacy-preserving face recognition. In *PETS*.

[14] Junfeng Fan and Frederik Vercauteren. 2012. Somewhat Practical Fully Homomorphic Encryption. *IACR Cryptology ePrint Archive* 2012 (2012), 144.

[15] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. 2011. Faster secure two-party computation using garbled circuits.. In *USENIX*. 331–335.

[16] Yan Huang, Lior Malka, David Evans, and Jonathan Katz. 2011. Efficient privacy-preserving biometric identification. In *NDSS*.

[17] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 604–613.

[18] Piotr Indyk and David Woodruff. 2006. Polylogarithmic private approximations and efficient matching. In *TCC*.

[19] Jianqiu Ji, Jianmin Li, Shuicheng Yan, Bo Zhang, and Qi Tian. 2012. Super-bit locality-sensitive hashing. In *NIPS*. 108–116.

[20] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. 2016. Efficient batched oblivious PRF with applications to private set intersection. In *CCS*.

[21] Margarita Osadchy, Benny Pinkas, Ayman Jarrous, and Boaz Moskovich. 2010. Scifi-a system for secure face identification. In *IEEE S&P*.

[22] Benny Pinkas, Thomas Schneider, Christian Weinert, and Udi Wieder. 2018. Efficient circuit-based PSI via cuckoo hashing. In *EUROCRYPT*.

[23] Maxim Raginsky and Svetlana Lazebnik. 2009. Locality-sensitive binary codes from shift-invariant kernels. *Advances in neural information processing systems* 22 (2009), 1509–1517.

[24] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. 2009. Efficient privacy-preserving face recognition. In *ICISC*.

[25] Ferdinando S Samaria and Andy C Harter. 1994. Parameterisation of a stochastic model for human face identification. In *IEEE WACV*.

[26] SEAL 2020. Microsoft SEAL (release 3.5). https://github.com/Microsoft/SEAL. Microsoft Research, Redmond, WA.

[27] Adi Shamir. 1979. How to share a secret. *Commun. ACM* (1979).

[28] Ebrahim M Songhori, Siam U Hussain, Ahmad-Reza Sadeghi, and Farinaz Koushanfar. 2015. Compacting privacy-preserving k-nearest neighbor search using logic synthesis. In *IEEE DAC*.

[29] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning.. In *AAAI*, Vol. 4. 12.

[30] The Guardian. 2020. https://www.theguardian.com/technology/2020/aug/11/south-wales-police-lose-landmark-facial-recognition-case.

[31] The Intercept. 2020. https://theintercept.com/2018/05/30/face-recognition-schools-school-shootings/.

[32] The NYT. 2020. https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html.

[33] The Verge. 2020. Moscow's facial recognition system can be hijacked. https://www.theverge.com/2020/11/11/21561018/moscows-facial-recognition-system-crime-bribe-stalking.

[34] Erkam Uzun, Simon P Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee. 2021. Fuzzy Labeled Private Set Intersection with Applications to Private Real-Time Biometric Search. In *30th {USENIX} Security Symposium*. 911–928.

[35] Erkam Uzun, Carter Yagemann, Simon Chung, Vladimir Kolesnikov, and Wenke Lee. 2021. Cryptographic key derivation from biometric inferences for remote authentication. In *ASIACCS*.

[36] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. 2016. EMP-toolkit. https://github.com/emp-toolkit.

[37] Masaya Yasuda. 2017. Secure Hamming distance computation for biometrics using ideal-lattice and ring-LWE homomorphic encryption. *Information Security Journal: A Global Perspective* 26, 2 (2017), 85–103.

[38] Qingsong Ye, Ron Steinfeld, Josef Pieprzyk, and Huaxiong Wang. 2009. Efficient fuzzy matching and intersection on private datasets. In *ISISC*.

[39] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503.