Giulia Fanti, Matthieu Finiasz, and Kannan Ramchandran

One-Way Private Media Search on Public Databases

[The role of signal processing]

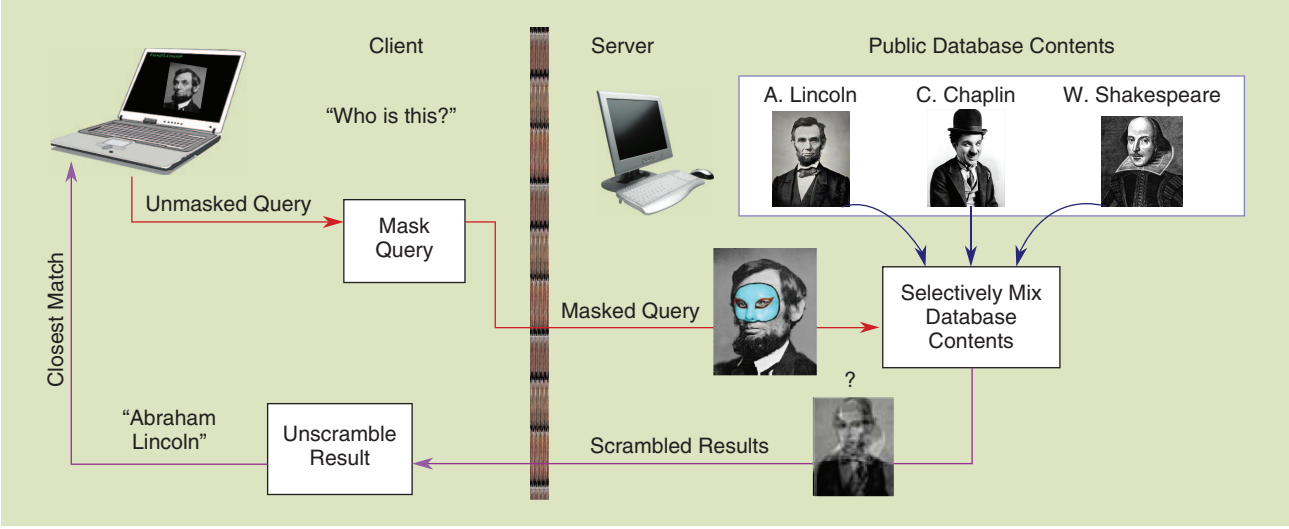
Automated media classification is becoming increasingly common in areas ranging from mobile location recognition to surveillance systems to automated annotation. While these tools can add great value to the public sphere, media searches often process private information; in such situations, it is important to protect the interests of one or both parties. Much attention has been given to the scenario where both the server and the client wish to keep their data secret, but comparatively little work has been done on searches in which only the client's data is sensitive. Nonetheless, there is great potential for applications involving private searches on public databases like Google Images, Flickr, or "Wanted Persons" directories put forth by various police agencies. In this article, we make the case that one-way private media search is an important and practically viable direction for future research. We will introduce readers to some basic one-way privacy tools and present a case study outlining the design of a private audio search tool on a public database. This case study serves as a backdrop for a discussion on the role of signal processing techniques in the design of privacy-preserving media search systems.

Digital Object Identifier 10.1109/MSP.2012.2229783
Date of publication: 12 February 2013

INTRODUCTION

The past few decades have introduced a multitude of applications that rely on the ability of devices to classify media, from voice transcription to face detection to optical character recognition. In a philosophical sense, the ability to classify stimuli is critical to learning and decision making; an animal will not flee a predator unless it is first able to identify the predator as such. However, while living creatures match stimuli to databases stored in their own brains, an automated recognition system must often match stimuli against an external database due to resource constraints. Inherent in this workload distribution is an implicit trust between client and server. If the client does not trust the server, is it still possible to achieve accurate and fast performance in a private setting? This question has become increasingly relevant in recent years, as researchers attempt to mitigate the costs of anonymity.

Explicitly, the inherent privacy problem is the following: If a system collects sensitive data for the purpose of recognition, it should avoid blindly submitting that data to an external database out of concern for the subject's or the client's own privacy. For example, consider an airport security system that matches the faces of travelers to an internationally maintained database of faces of suspected terrorists (e.g., Interpol's "Wanted Persons" database) [6]. The airport would like to ensure that travelers are not suspected terrorists, but at the same time, Interpol should



[FIG1] System diagram for a one-way private face-recognition system. The client wishes to classify a media file in such a way that the server cannot learn information about the client's query. Red arrows signify data provided by the client, blue arrows by the server, and purple by a combination thereof.

not receive any information about the travelers whose faces do not appear in their database. Thus, the notion of private media content recognition is extremely appealing. Although secure biometric authentication is the main application typically cited for this research [1], [6], there are many other possible uses, such as private image location estimation [7] or speech processing [27].

Traditional media recognition techniques are difficult to adapt to the private domain for several reasons. The high computation and communication costs associated with privacy primitives constrain the size of media feature spaces in practice, but smaller features can also reduce the accuracy of search algorithms. Moreover, cryptography tools are typically exact and intolerant of any loss in representation, while media objects are inherently distortion tolerant; effective classification algorithms tend to exploit this tolerance for performance gains. Reconciling these conflicting characteristics can make encrypted-domain media content recognition challenging. Nonetheless, the area has received significant attention as researchers overcome technical limitations by exploring new mathematical tools and cleverly applying existing mathematical tools.

There are two main classes of problems that deal with the described privacy issue. The first class is more stringent and assumes that neither the client nor the server is willing to sacrifice any information, which applies when the server stores sensitive data like medical records or classified information. In the context of media searches, this problem has received significant attention in recent years with promising results in image and speech processing [1], [6], [11], [25], [27]. However, two-way private media search systems can be heavy in computation and/or communication due to the fact that they have very strong privacy demands.

The second class of problems assumes that only the client wishes to protect his/her data, which applies whenever the database is public. This latter class will be the focus of our discussion. To clarify, a sample diagram of such a system is shown in Figure 1. We see here the information flow in a one-way private face-recognition system that aims to identify an image of Abraham Lincoln. At a high level, the design problem can be reduced to specifying three main factors: query masking, query processing, and result deciphering.

This second class of user-private media searches has received comparatively little attention in the literature; the application space is nonetheless ample. Consider, for instance, the wealth of information stored on Google Images or Flickr that could be used for face or location recognition. Alternatively, in the airport example given earlier, the database of suspects is likely to be public; many police agencies publish the images of wanted or missing persons on their Web sites. In the commercial sphere, privacy-preserving recommendation systems could improve users' shopping experiences without revealing their preferences to companies that are likely to sell that information. One could envision services that automatically classify medical data (e.g., DNA sequences, electrocardiography signals) by comparing against public databases of classification parameters; such tools could be useful in areas where medical access is sparse.

The objective of this article is to argue that one-way-private content-based media classification is both inevitable and feasible. In the interest of readability, we provide a broad view of relevant ideas and refer the reader to the literature for more detailed explanations. We begin by briefly explaining the development of private media searches over the past decade. This is followed by a description of asymmetric privacy primitives and a design example in the form of a user-private music search tool. This tool exemplifies a media recognition algorithm that is easy to adapt to the private domain, so we will then use similar ideas to explore design principles in one-way-private media search systems. An important message of this article is that pairing privacy primitives with existing recognition algorithms is rarely a practical solution in this space because of resulting communication and computation costs. Rather, efficient private search systems will typically involve

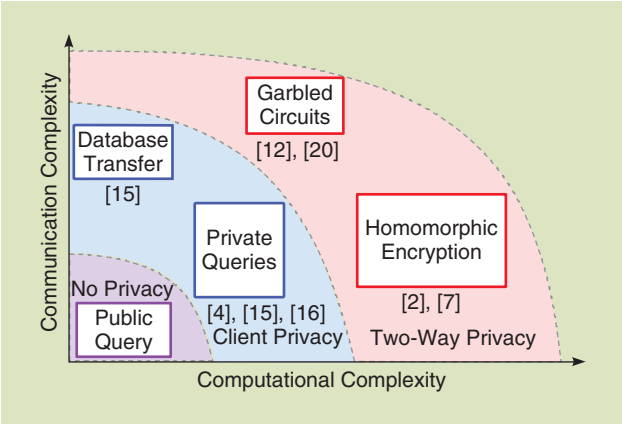
deliberate modifications to existing media recognition algorithms. We conclude by going over some major challenges that have yet to be overcome in one-way private media searches.

CURRENT SYSTEMS

Private searches place tighter demands on a search algorithm than public searches do, and they therefore usually cost more in resources like communication and computation. Although the notion of private computing is garnering significant public interest, users are ultimately unlikely to tolerate the associated drawbacks if those drawbacks manifest themselves as significant delay and/or increased communication charges. Thus the primary goal of most private media matching research is to reduce the total resource demands of private search systems [1], [11], [12], [25].

Figure 2 depicts the relative cost profiles of some existing private media search techniques. In practice, algorithms often combine several privacy primitives, so the schematic is meant only as a qualitative guide to the distribution and magnitude of costs for each primitive. The majority of existing schemes belong to the outer layer of the figure, i.e., they are two-way private approaches. The relations between two-way primitives will be covered in greater depth in other articles, but one general trend emerges: there is usually a tradeoff between required communication and computation. For instance, consider algorithms that rely on homomorphic encryption [1], [6]. In these algorithms, the client sends an encrypted query to a server, and due to mathematical properties of the cryptosystem, the server is able to do computations on the query despite not knowing its content. Such algorithms are typically computationally heavy (compared to a public search), but the communication costs can be fairly low. On the other hand, there are schemes that rely on garbled circuits in which the server effectively sends a garbled version of the database to the client; the client then extracts a single desired element [11], [25]. These schemes require far less computation, but the communication costs can be quite high for most practical circuit sizes. It is possible to mitigate this cost with locally generated hardware tokens [13], but the server must trust the token, which is susceptible to tampering in practice. Researchers have drastically lessened the individual limitations of these primitives by combining them in clever ways [2], [25].

The middle band in Figure 2 indicates primitives guaranteeing only one-way privacy. These tools can achieve desired privacy levels at lower computation and communication cost than two-way schemes. This comparison is not really fair because the two classes solve different problems, but the point is that there is a practical reason to treat the two cases differently. One-way privacy tools have received very little attention in the context of media searches. To our knowledge, Shashank et al. are the only researchers to build a system based on one-way private media matching, in the form of a hierarchical, user-private image similarity search tool [12]. Others have proposed ways of improving the efficiency of two-way-private schemes when databases are partially or completely public [2], [6], [17]. Yet, more than ever, the current technological and social landscape is conducive to user-private media recognition and will presumably continue to evolve in this



[FIG2] Media search techniques categorized in terms of privacy level, computational efficiency, and communication requirements.

direction. On the cryptographic side, the required tools have only recently matured enough to be considered viable. Practically speaking, one-way privacy primitives were long thought to be as inefficient as transmitting the server's entire database to the client [26]. The issue was not formally addressed until 2011, when Olumofin and Goldberg showed that private queries can actually be orders of magnitude more efficient than database transmission in practice [16]. In fact, developments in one-way privacy primitives have rendered costs sublinear in both communication and computation under certain conditions [3], [5], [29].

On the other hand, public media databases have become commonplace only within the last decade (e.g., YouTube, Flickr, Facebook, Google Images), and these services did not introduce similarity searches until even more recently. The tremendous growth of public databases has proved increasingly useful for media matching applications and also heightened the potential privacy threat to the average user [7]. Indeed, within the last year, both Google and Facebook have had to settle lawsuits with the Federal Trade Commission for violating the privacy of users [22]. In short, the issue of user privacy is a very relevant one, both from the client's and the server's perspective. The combination of these factors with cryptographic advances suggest that user-private media searches can and should receive increased attention in coming years.

PRIVATE QUERIES

We will start by explaining the fundamentals of one-way-private cryptographic tools, collectively termed "private queries." Private queries are searches on a database in which both the query and the result are masked from the server. The caveats are twofold: the client must know exactly what data is desired from the database, and the client may learn information about the database besides just the desired file during the search process. The latter condition is unimportant on public databases, but the fact that the user must know precisely what to request is problematic since media searches are inherently inexact. We will discuss ways to get around this stringent requirement in the section "Case Study: Private Audio Recognition System." The current section describes two

representative private query methods used to privately retrieve items from a database: one taken from [5] that uses multiple servers to achieve information-theoretic security, and one based on [19] that uses a single server to provide computational security.

MULTISERVER PRIVATE INFORMATION RETRIEVAL

Private information retrieval (PIR) allows a client to retrieve data at a particular index in a database without revealing the query (or the results) to the server. As mentioned above, there exist PIR implementations that require only one server [14], [18], though in practice they are significantly less efficient than trivial database transfer [26]. We will emphasize multiserver schemes, which can achieve communication and computation that is sublinear in database size [3], [29]. Multiserver PIR schemes require the existence of at least two noncolluding servers, each with a duplicate copy of the database. This assumption is strong, but not unreasonable; for instance, one could store data on clouds run by competing services, e.g., Amazon and Google.

We first describe the most basic PIR scheme from [5]; while the communication complexity is order-equivalent to database transfer, it is a useful educational example and can be generalized for efficiency gains. Both servers possess copies of a database comprising a binary string $x \in \{0,1\}^n$, and the user wishes to retrieve the i th bit, x_i. The user's request can be represented by $e_i \in \{0,1\}^n$, the indicator vector with a one at index i and zeros elsewhere. To disguise this query, the user generates a random string $a \in \{0,1\}^n$ with each entry a Bernoulli (1/2) random variable. The queries sent to Servers 1 and 2 are $a \oplus e_i$ and a, respectively. Each server computes the inner product of its received query vector with the database x using bitwise addition (exclusive or, denoted XOR hereafter) and returns a single-bit result. The user XORs the results from the two servers to get precisely x_i. The scheme is illustrated in Figure 3; if the database consists of indexed files rather than bits, the same process is carried out on each bit plane.

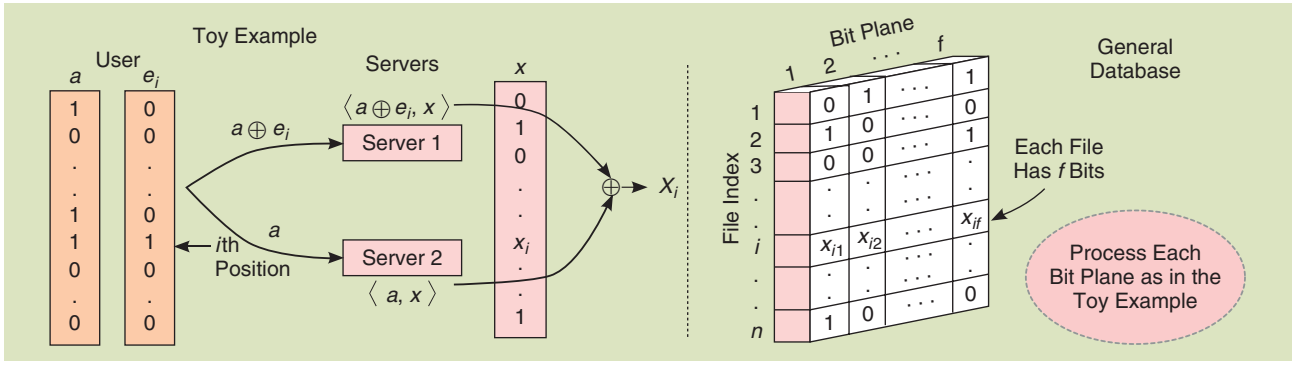
This multiserver PIR scheme is information-theoretically secure, meaning that an adversary cannot break the scheme even with unlimited computing power. Among the PIR schemes that require communication linear in database size, this multiple-server version is the lightest because each bit in the database is simply XORed once, rather than being processed with weightier cryptographic operations. However, by embedding the database in a d-dimensional cube, where $d \geq 2$ is the number of servers, the communication can be further reduced to $O(\sqrt[d]{n})$, or $O(\sqrt{n})$ in our two-server setup [5]. This simple extension of the basic PIR scheme generalizes the idea for multiple servers. Meanwhile, other techniques like precomputation have been proposed to reduce the computational requirements while maintaining efficient communication [3], [29].

SINGLE-SERVER PIR: PRIVATE STREAM SEARCH

As mentioned earlier, PIR can be done with a single server at the expense of computation and communication [14]; the resulting level of security is computational rather than information-theoretic, so the scheme cannot be broken by an adversary with bounded computational resources. We now present such a single-server computationally private PIR scheme. In addition to our one-way privacy application, the tools used in this scheme have mathematical properties suitable for implementations of two-way privacy that appear elsewhere in this issue of *IEEE Signal Processing Magazine*.

Private stream search (PSS) was originally designed to perform private keyword searches on databases [19]. For instance, one might want to privately retrieve all files in a text document database that contain the word "red." The problem statement is a bit different from that of a PIR query, since the client is no longer searching for a specific file by index. However, this setup can easily be framed as PIR by making the dictionary of keywords equal the list of document indices. We will explain the scheme from [19] in the context of PIR for ease of comprehension, but the original framing of PSS is also useful and represents a different variety of private query.

This version of PIR relies on additively homomorphic encryption, a technique utilized to varying degrees in almost every current private media content retrieval scheme. In the general case, a public-key cryptosystem composed of encryption function $\mathcal{E}(\cdot)$ and decryption function $D(\cdot)$ is homomorphic if there exist operations $f_1(\cdot)$ and $f_2(\cdot)$ such that $f_1(x, y) = D(f_2(\mathcal{E}(x), \mathcal{E}(y)))$ [24]. There are several cryptosystems with this property, but the additively homomorphic Paillier cryptosystem is used in the majority of existing private media content retrieval applications [20]; it is



[FIG3] Basic PIR scheme. Each of the two servers computes the bitwise sum of a user-specified subset of database files. $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. Because the two user-specified subsets differ only at the *i*th index, the binary addition of the results from both servers gives the original desired file.

homomorphic with f_1 corresponding to addition and f_2 to multiplication, i.e.,

$$\mathcal{E}(x+y) = \mathcal{E}(x)\,\mathcal{E}(y). \tag{1}$$

When x and y are scalar integers, this implies that multiplication by a constant c takes a particularly simple form

$$\mathcal{E}(cx) = \mathcal{E}(x)^c. \tag{2}$$

One important point is that the Paillier cryptosystem has the property of semantic security, which means that a computationally bounded adversary—in this case, the server—cannot derive significant information about an encrypted message given only the encrypted message and the public encryption key. This definition implies that the encryption scheme must be random, so successive encryptions of the same message cannot be detected as such with high probability.

Now we apply this to the single-server PIR problem. Consider an ordered list of possible file indices, from one to n, with i being the desired index. Using an additively homomorphic cryptosystem, the client generates an encrypted query vector q of length n, with $\mathcal{E}(1)$ at index i and $\mathcal{E}(0)$ at the remaining indices. Because the Paillier cryptosystem is randomized, all the entries in this encrypted vector will be different with high probability. The encrypted query vector is sent to the server, as shown in Figure 4. Now the server goes through its database; for every file index, the server checks the corresponding entry in the encrypted query vector. This entry, which is either $\mathcal{E}(0)$ or $\mathcal{E}(1)$, is raised to the power of the whole file represented as a number. So if $i = 1$, then $q_1 = \mathcal{E}(1)$, giving $\mathcal{E}(1)^{f_1}$; by (2), this quantity is equivalent to the encryption of f_1. The same procedure is also done for all the other files f_j, $j \neq i$, but in that case, $\mathcal{E}(0)^{f_j} = \mathcal{E}(0)$. After going through the whole database, the results of these exponentiations are multiplied together and returned to the client. Because of the homomorphic cryptosystem, multiplying ciphertexts corresponds to adding their arguments, so we get $\mathcal{E}(0 + \cdots + 0 + f_i + 0 + \cdots + 0) = \mathcal{E}(f_i)$. Thus the client decrypts precisely the file at the desired index. A more detailed explanation of the technique is provided in [19], albeit in the context of a keyword search.

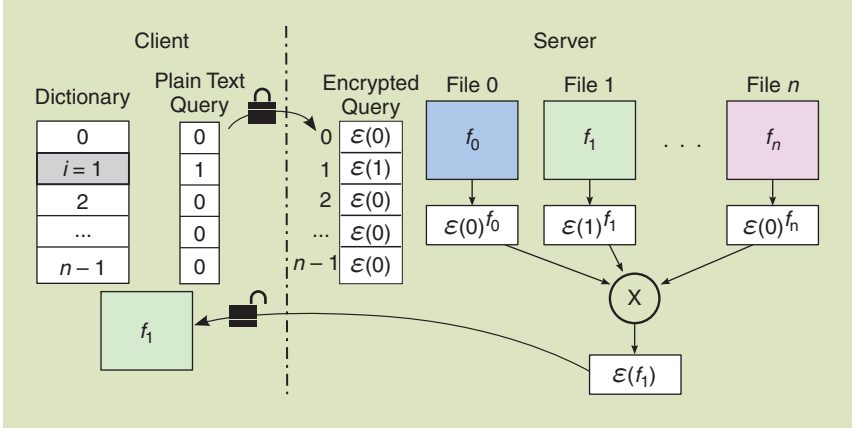
CASE STUDY: PRIVATE AUDIO RECOGNITION SYSTEM

As mentioned in "Introduction" section, a practically viable private media search system will typically feature a customized search algorithm. In the early days of privacy-preserving signal processing research, the focus was mostly on pairing cryptography techniques with existing search algorithms [6]. This approach can certainly add unnecessary costs, but there do exist accurate search algorithms that are already compatible with private searches. We will start by presenting a private audio search tool based on precisely such an algorithm, i.e., the audio search tool of Haitsma and Kalker [10]. The basic premise is as follows: A mobile client has a sound clip recorded from a noisy source and wishes to learn the name of the song from a remote server without revealing anything about the noisy clip to the server. The privacy constraints here are contrived, since there is usually no reason to mask one's musical preferences, but this example is meant as a stepping stone to other media forms like images and video.

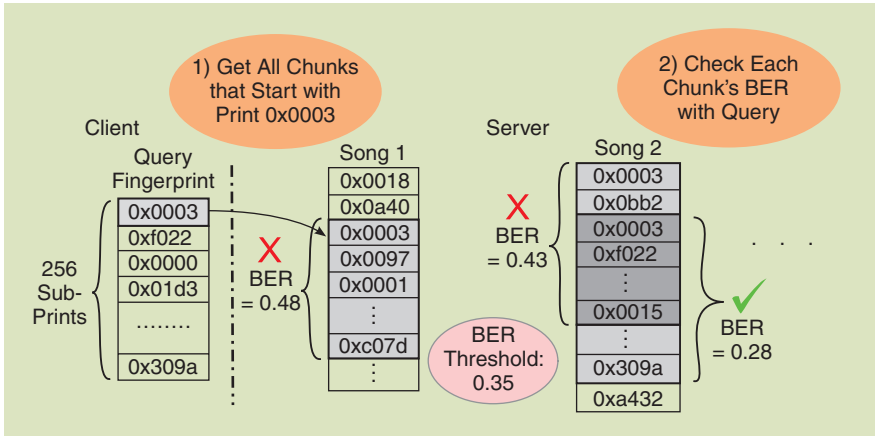
An important limitation of the privacy techniques described earlier is that the client must always know the precise search index. However, media searches consist of finding similar entries, not identical ones; the query will rarely fully match a database entry. One way to get around this problem is to find feature vectors in the database that are similar in a holistic sense to a query feature vector. This is the approach taken by the majority of privacy-preserving media recognition systems [2], [6], [12], [17]. A different solution, inherent to Haitsma's and Kalker's scheme, is to conduct an exact search for a broader group of entries that probably includes the desired content, and then select the closest entry within that set. That is, suppose "Equinox" by John Coltrane comes on the radio, and a listener wants to learn the song title without revealing information to the server. The listener does not know exactly which song to search for, but he/she will most likely recognize the music genre as jazz. A private search for all jazz music will return the desired song (among many others), while weeding out a majority of possible files. This example is coarser than the actual implementation, but the underlying idea is the same. We will start by giving a brief overview of the nonprivate audio search scheme before illustrating how one could easily adapt it to the private domain; detailed descriptions can be found in [10].

ORIGINAL ALGORITHM

Each audio file in the database is represented as a collection of time-dependent, quantized audio features called subfingerprints. The features describe the frequency-domain content over



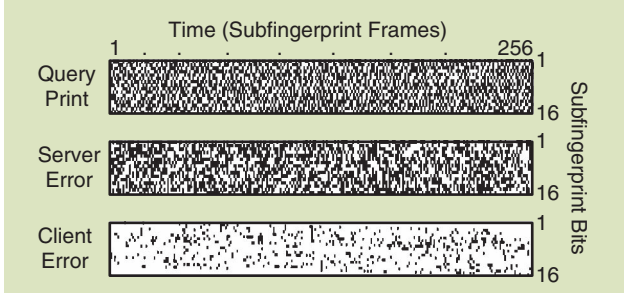
[FIG4] Example of a PSS search in which the user wants the *i*th file from a database. The server, which only has the public encryption key, computes *n* modular exponentiations and returns the product of all of these. The client then uses the private key to decrypt the results and obtain the desired file.



[FIG5] Diagram of public audio search algorithm. The client successively sends each subfingerprint to the server, which returns all chunks of 256 prints that match the query exactly at the specified position. The client checks the bit error rate (BER) between the query and the returned chunk and declares a match if the BER is below a threshold.

windowed time frames, and quantization of these features allows different, noisy renditions of the same clip to map to the same 16-b subfingerprint, thereby adding robustness. The longer a song is, the more subfingerprints are required to describe it. We will describe a slightly modified version of the original algorithm to facilitate the transition to the private domain.

On the client end, a 3-s query audio clip, potentially corrupted by noise or other distortions, is also converted to a subfingerprint representation—3 s of audio map to 256 subfingerprints, which are collectively termed a "fingerprint block." The user sends the 256 noisy query subfingerprints to the server; for each one, the server checks all fingerprint blocks in the database that contain the desired value at the desired position. For example, if the first subfingerprint has the value 0x0003, then the algorithm will find all fingerprint blocks that start with the value 0x0003. If one of those blocks is very similar to the query in Hamming distance, a match is declared. Otherwise, the server checks the second query subfingerprint, and so forth until it has tried all 256 query prints. This scheme is illustrated in Figure 5. Note that each time we search for the ith of the 256 query subfingerprints, the server checks all chunks in the database that exactly match the query prints in the ith location; this property sets the stage for an easy transition to the private domain.



[FIG6] "Query print" shows the fingerprint block for a noisy query song (SNR = 15). "Server error" shows the difference between one PIR server's output and the query block. Bits in error are shown in black. "Client error" shows the difference between the client's unscrambled result and the original.

PRIVATE VERSION

To make the scheme private, we convert every query subfingerprint search to a PIR query, thereby hiding all information from the server. In our analysis, we will assume an underlying multiserver PIR scheme, since single-server cryptographic schemes are unlikely to gain traction as practical privacy tools. Computationally private schemes require the use of asymmetric encryption operations which, when used with secure key sizes, are very expensive. In practice, information-theoretically secure PIR can be at least 1,000 times faster.

The client begins by converting the 3-s noisy query to a list of 256 subfingerprints, just as before. Now the client should (privately) send the subfingerprints to the server to be matched. Suppose again that the first subfingerprint takes the value 0x0003—however, the client does not know which locations in the database hold subfingerprint 0x0003. By giving the database an inverted structure (i.e., indexing it by subfingerprint value), we can frame the problem as a private query. Now the client only needs to submit a PIR query for index 0x0003, after which the server returns all fingerprint blocks containing subfingerprint 0x0003. Then the client can compare the BER on the returned fingerprints to a prespecified threshold. The ith index in the database holds all blocks that contain subfingerprint i rather than just those that start with i. This is because if the first query subfingerprint q_0 does not result in a match, we move on to the second subfingerprint q_1. Then we seek all blocks that contain q_1 in the second position, and so forth. This is not the only solution, but it is a straightforward and relatively communication efficient one.

PERFORMANCE

The recognition rate of the algorithm is completely independent of the privacy settings because PIR deterministically returns the desired file. Ultimately, the clip can only be identified if there is at least one exact match in the query fingerprint block. Preliminary tests on a random set of 100 popular songs showed that at a typical FM radio signal-to-noise ratio (SNR) of 70 dB, this happens 98% of the time. For a qualitative notion of the information obtained by the server compared to that obtained by the client, Figure 6 gives a binary view of various fingerprint blocks. The query block is highly correlated with the client's results, so the difference of the two yields very few bit errors, while the difference between the server's output and the query block appears to be totally random. Indeed, since the PIR scheme is information-theoretically secure, the server learns nothing from a subfingerprint query. A demonstration of the audio search system in action can be found on http://www.eecs.berkeley.edu/~gfanti/demo_audio.html.

In its basic form, our scheme has a worst-case (and average) communication complexity of $\mathcal{O}(\max(p(k), m(n))$, where n is

the database size and $m(n)$ is the expected number of exact subfingerprint matches in the database, k is the number of bits in each subfingerprint, and $p(k)$ is the total communication complexity of a PIR search on a list of k-b subfingerprints. For good discriminative power and search efficiency, we assume that k is chosen according to the database size; if k is chosen as $\mathcal{O}(\log(n))$, then the expected number of matches will scale as $\mathcal{O}(1)$ with the database size, and the dominant communication cost will come from the uplink PIR queries. This is good, because it means that efficient PIR schemes can reduce the communication to sublinear levels.

For the computation costs, increasingly large private queries (i.e., increasingly large numbers of bits per subfingerprint) cause the server-side computation to dominate. Servers are better suited than clients to handle heavy computation, particularly since these tasks can be easily parallelized. Also, private query computation can be reduced with better PIR schemes like [3] and [29], while portions of the algorithm unrelated to privacy cannot be trimmed as easily. Table 1 shows a comparison of communication and computation costs in this audio search scheme using various PIR varieties. An important point is that although we can achieve sublinear communication and computation, the best known PIR schemes have polynomial communication in n, and thus cannot match the logarithmic costs of nonprivate communication with any number of servers.

DESIGN CONSIDERATIONS

The Haitsma and Kalker search algorithm is a useful example because only minimal changes are required to make the algorithm privacy-preserving. This situation is by no means typical, and good systems will usually require special search algorithms offering both private-domain compatibility and resource efficiency. Designing such algorithms involves identifying low-dimensional features that are conducive to exact comparison, changing actual search mechanisms, or both. We will start by discussing some feature vector modifications that can make media search algorithms easier in the private domain; this is followed by a sample face recognition tool that relies on these techniques and also meshes nicely with the audio search tool just presented. We will then touch upon ways of modifying search algorithms for private search.

Many media recognition algorithms search for Euclidean-distance nearest-neighbor vectors in a database [6], [11], [25]. Exact matches are convenient for the private query aspect of the algorithm, but it is intuitively clear that nearest-neighbor searches will generally give better recognition results. A popular research topic in recent years aims to represent arbitrary feature vectors in a way that reduces nearest-neighbor searches to a Hamming distance comparison, which is computationally lighter in both the private and nonprivate domain. A lot of this work relies on locality-sensitive hashing techniques that provide probabilistic noise resistance. For instance, [30] and [15] have explored a simple but effective hashing technique: project both vectors onto a set of random hyperplanes and record the sign (+/−) of each projection for each vector. It turns out that the Hamming distance of these binary hash vectors is a probabilistic estimate of Euclidean distance. Indeed, Min et al. found this technique to give 95% accurate nearest-neighbor estimates at a fraction of the cost for high-dimensional features like gist [15]. Other less-studied feature modifications may be introduced specifically for a given application; examples include the quantization techniques used in [2] and distance-preserving hashing in [21]. We used examples from the two-way privacy literature since there are not many one-way private systems in existence.

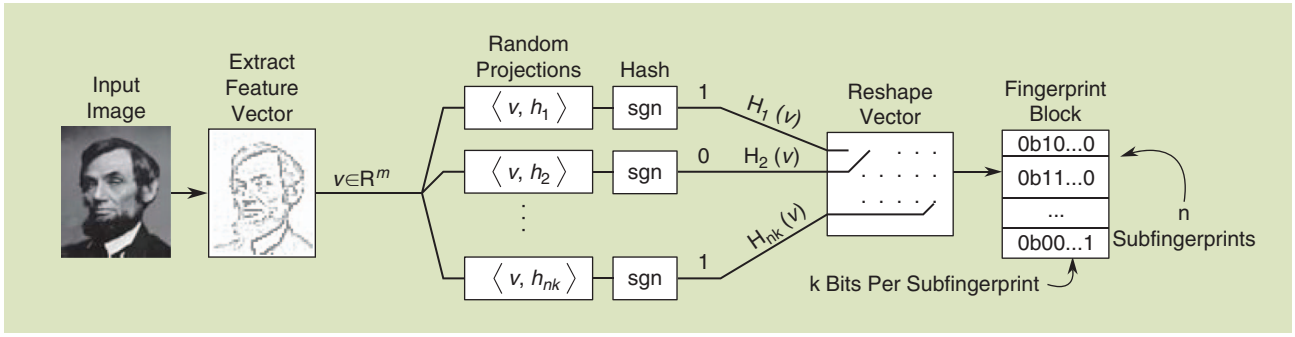
To give a simple example of how feature vector alteration can be useful for one-way privacy, suppose we are identifying facial images with the Eigenfaces algorithm [28]. All images in the system have an associated feature vector, obtained through projection onto a basis of so-called "Eigenfaces"; the details of feature vector extraction are not important here. The original search algorithm consists of finding the closest database feature vector in Euclidean distance to the normalized query feature vector. Using the locality-sensitive hashing technique explained above, we can hash all these feature vectors to obtain a sequence of bits that can simply be grouped into "subfingerprints" as before. It is then straightforward to apply Haitsma and Kalker's algorithm to the private face recognition problem (or any other problem involving feature vector nearest neighbor matching). Figure 7 illustrates how to modify an arbitrary feature vector for compatibility with the exact-match search scheme of [10]. We implemented a face recognition system based on this approach, and a demonstration video can be seen at http://www.eecs.berkeley.edu/~gfanti/demo_face.html. Table 2 gives asymptotic communication and computation costs for this face recognition scheme as well as two benchmark two-way privacy schemes [17], [25]. Again, the comparison is not fair since the two-way private schemes solve a fundamentally harder problem, but we wish to highlight that order-level gains can be had by exploiting one-way privacy. In addition, the system for secure computation of face identification (SCiFI) also has much higher accuracy than the Eigenfaces algorithm.

A complementary tactic is to modify or specifically design search algorithms for private search. This is done to varying degrees in most privacy-preserving media search systems by altering overarching factors like database structure or search algorithms [6], [11], [12], [25]. In the context of privacy-conscious design, there are also some seemingly secondary issues that nonetheless require attention. One such issue is feature size—in the nonprivate domain, there are ways to handle the comparison of

[TABLE 1] ASYMPTOTIC COMMUNICATION AND COMPUTATION COSTS FOR PRIVACY-PRESERVING AND NONPRIVATE AUDIO SEARCH.

PRIVACY SCHEME	COMMUNICATION	COMPUTATION
TWO-SERVER PIR, SEC. 3.1	$O(\sqrt{n})$	$O(n)$
TWO-SERVER PIR [3]	$O(\sqrt[3]{n})$	$O\left(\frac{n}{\log^2 n}\right)$
SINGLE-SERVER PIR [14]	$O(n^c)$, FOR ANY $c > 0$	$O(n)$
NONPRIVATE [10]	$O(\log n)$	$O(\log n)$

n is database size and k is number of bits per subfingerprint. For consistency, we choose $k = \log n$ in both the private and nonprivate audio search schemes.



[FIG7] Method for converting arbitrary feature vectors into "subfingerprints." This facilitates a private search similar to the one used in the previously described audio search. h_i denotes a randomly drawn m-dimensional hyperplane, where m is the original feature vector length.

differently-sized features, but these methods can rely on computationally heavy metrics like the earth mover's distance or Levenshtein distance. These techniques are impractical in the private domain, so it's important to ensure that the feature space is directly comparable between the database and the queries, as with the subfingerprints in the audio search scheme. Another issue is alignment—query media files collected in an uncontrolled environment are likely not aligned in the same way as the database files. Once again, standard alignment algorithms are difficult to execute in the private domain, so the issue merits special attention. An excellent example of a scheme that addresses all these issues and was fully designed for the private domain is the SCiFI system by Osadchy et al. [17]. By defining faces as a collection of items from dictionaries of face parts and part locations, the authors facilitate comparison in the encrypted domain. Specifically, the recognition step is defined by whether a set difference (a Hamming distance calculation) is above or below a threshold. This kind of search algorithm overhaul can lead to better recognition rates and fewer unnecessary algorithm inefficiencies.

CHALLENGES AND FUTURE DIRECTIONS

One-way privacy techniques are still a long way away from public use, due in large part to the inefficiency of known one-way privacy techniques. However, as media recognition becomes a mainstream feature, this increasingly relevant area of study is likely to gain significance. The major remaining issues are twofold. On the cryptographic front, there are a few research areas that would facilitate the commercialization of one-way private queries. One of the most appealing is the development of fully homomorphic encryption schemes. Fully homomorphic schemes would permit computation of encrypted domain functions involving multiplications and additions, rather than just one operation [8]. This option is currently far from viable due to efficiency limitations, but it is nonetheless an active field. There is also ongoing work on improving classical private queries through techniques like preprocessing [3]. In a completely different vein, recent research on quantum private queries has suggested as much as an exponential reduction in computational and communication efficiency with respect to existing classical implementations [9]. Quantum private queries are still in the very early stages of development, but their realization would impact privacy-preserving tactics immensely.

An important point is that feature vectors, whether hashed or not, should not be sent directly to the database because they contain information about the original query. In the face recognition schemes, for instance, we perfectly masked the hash bits in the form of a PIR query, just like in the audio search of the section "Case Study: Private Audio Recognition System." However, it would be even better if we could do away with privacy primitives entirely and instead rely on privacy-inherent feature representations. To this end, some researchers are trying to build methods for giving feature vectors privacy properties without the bulk of cryptography techniques. For instance, [23] showed that compressed sensing measurements offer both dimensionality reduction and computationally secure privacy. Similarly, works like [4] and [21] have used quantized and masked feature vectors in two-way private scenarios to significantly reduce communication costs while providing privacy in different scenarios. It is not clear yet whether such techniques will have a place in privacy-preserving media searches on public databases, but the idea is certainly appealing from a number of perspectives.

The relevance of these componentwise advances will ultimately depend on the availability of signal processing algorithms that are able to integrate them. In the absence of good universal media search algorithms, we must focus on finding representations of data and associated search schemes that give minimal accuracy loss and efficient operation in the private domain. For instance, nearest-neighbor approaches can be used on arbitrary media feature vectors and can be recast for the one-way private domain with locality-sensitive

[TABLE 2] FACE RECOGNITION ASYMPTOTIC COMMUNICATION AND COMPUTATION COSTS FOR PRIVACY-PRESERVING FACE RECOGNITION.

ALGORITHM	PRIVACY SCHEME	COMMUNICATION	COMPUTATION
HASHED EIGENFACES	TWO-SERVER PIR [3]	$O(\sqrt[3]{n})$	$O\left(\frac{n}{\log^2 n}\right)$
EIGENFACES [25]	HOMOMORPHIC ENC., GARBLED CIRCUITS	$O(n)$	$O(n)$
SCiFI [17]	OBLIVIOUS TRANSFER	$O(n)$	$O(n)$

"Hashed Eigenfaces" refers to our suggested scheme, and we also compare this to the online complexities of two benchmark face recognition systems [17], [25]. We chose $k = \log n$ in the hashed scheme, where k is subfingerprint length and n is the number of faces in the database.

hashing methods. The goal is to find analogous conversion schemes with high accuracy, applicability, and costs sublinear in database size. Meanwhile, many of the most successful media-matching algorithms are computationally complex and require tools that are currently too heavy for the encrypted domain. If the engineering community wishes to pursue private media search as a viable technology—and there is strong incentive to do so—then sufficiently accurate classification algorithms must be developed that can easily be adapted to the private domain. Whether enabled by cryptographic advances or signal processing ones, this compatibility requirement will play a large role in determining what kinds of applications can ultimately be supported in a private setting.

ACKNOWLEDGMENTS

This material is based on work supported by National Science Foundation (NSF) grant CCF-0964018 and by the NSF Graduate Research Fellowship grant DGE-1106400.

AUTHORS

Giulia Fanti (gfanti@eecs.berkeley.edu) is an electrical engineering M.S./Ph.D. student in the Wireless Foundations group at the University of California at Berkeley. She received her B.S. degree in electrical and computer engineering from Olin College of Engineering in 2010.

Matthieu Finiasz (finiasz@gmail.com) did his Ph.D. work at INRIA (Paris, France) from 2001 to 2004 and was a postdoctoral researcher at EPFL (Lausanne, Switzerland) from 2004 to 2007. After five years as a research scientist at ENSTA (Paris, France) he joined the CryptoExperts company in 2012. His research interests are the design of efficient code-based cryptosystems, applications of coding theory to symmetric and asymmetric cryptography, and, more generally, most topics related to cryptography, anonymity, and privacy protection.

Kannan Ramchandran (kannanr@eecs.berkeley.edu) has been a professor in the Electrical Engineering and Computer Science Department at the University of California at Berkeley since 1999. Prior to that he was a faculty member at the University of Illinois at Urbana-Champaign from 1993 to 1999. He is a Fellow of the IEEE. His research interests include image and video compression and transmission, distributed signal processing, distributed storage, multiuser information theory, and multimedia networking.

REFERENCES

[1] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, A. Piva, and F. Scotti, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *Proc. IEEE Int. Conf. Biometrics: Theory Applications and Systems*, Washington, D.C., 2010, pp. 1–7.

[2] M. Barni, P. Failla, R. Lazzeretti, A. R. Sadeghi, and T. Schneider, "Privacy-preserving ECG classification with branching programs and neural networks," *IEEE Trans. Inform. Forensics Sec.*, vol. 6, no. 2, pp. 452–468, 2011.

[3] A. Beimel, Y. Ishai, and T. Malkin, "Reducing the servers' computation in private information retrieval: PIR with preprocessing," *J. Cryptol.*, vol. 17, no. 2, pp. 125–151, 2004.

[4] P. Boufounos and S. Rane, "Secure binary embeddings for privacy preserving nearest neighbors," in *Proc. Int. Workshop on Forensics and Security,* Guacu Falls, Brazil, 2011.

[5] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. IEEE Symp. Foundations of Computer Science,* Milwaukee, WI, 1995, pp. 41–50.

[6] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *Privacy Enhancing Technologies* (Lecture Notes in Computer Science, vol. 5672), I. Goldberg and M. Atallah, Eds. Berlin/Heidelberg, Germany: Springer**,** 2009, pp. 235–253.

[7] G. Friedland and R. Sommer, "Cybercasing the joint: Privacy implications of geo-tagging," in *Proc. USENIX Workshop on Hot Topics in Security,* Washington, D.C., 2010.

[8] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., 2009.

[9] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum private queries: Security analysis," *IEEE Trans. Inform. Theory, vol. 56,* no. 7, pp. 3465–3477, July 2010.

[10] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proc. Int. Symp. Music Information Retrieval*, Paris, France, 2002, pp. 107–115.

[11] Y. Huang, L. Malka, D. Evans, and J. Katz, "Efficient privacy-preserving biometric identification," in *Proc. Network and Distributed System Security Symposium*, San Diego, CA, 2011.

[12] J. Shashank, P. Kowshik, K. Srinathan, and C. V. Jawahar, "Private content based image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* Anchorage, AK, 2008, pp. 1–8.

[13] K. Järvinen, V. Kolesnikov, A. R. Sadeghi, and T. Schneider, "Embedded SFE: Offloading server and network using hardware tokens," in *Financial Cryptography and Data Security* (Lecture Notes in Computer Science, vol. 6052). Berlin/Heidelberg,Germany: Springer, 2010, pp. 207–221.

[14] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," in *Proc. Annu. IEEE Symp. Foundations of Computer Science*, New Brunswick, NJ, 2012, pp. 364–373.

[15] K. Min, L. Yang, J. Wright, L. Wu, X. S. Hua, and Y. Ma, "Compact projection: Simple and efficient near neighbor search with practical memory requirements," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, 2010, pp. 3477–3484.

[16] F. Olumofin and I. Goldberg, "Revisiting the computational practicality of private information retrieval," in *Financial Cryptography and Data Security* (Lecture Notes in Computer Science, vol. 7035), G. Danezis, Ed. 2012, pp. 158–172.

[17] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "SCiFI: A system for secure face identification," in *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, 2010, pp. 239–254.

[18] R. Ostrovsky and W. Skeith, "A survey of single-database private information retrieval: Techniques and applications," in *Public Key Cryptography—PKC 2007* (Lecture Notes in Computer Science, vol. 4450), T. Okamoto and X. Wang, Eds. Berlin/Heidelberg,Germany: Springer, 2007, pp. 393–411.

[19] R. Ostrovsky, W. Skeith, and O. Patashnik, "Private searching on streaming data," *J. Cryptol.*, vol. 20, no. 4, pp. 397–430, 2007.

[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT '99* (Lecture Notes in Computer Science, vol. 1592), S. Jacques, Ed. Berlin/Heidelberg,Germany: Springer, 1999, pp. 223–238.

[21] M. Pathak and B. Raj, "Privacy-preserving speaker verification as password matching," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Kyoto, Japan, 2012, pp. 1849–1852.

[22] J. Pepitone. (2011, Nov. 10). Facebook settles FTC charges over 2009 privacy breaches. CNN [Online]. Available: http://money.cnn.com/2011/11/29/technology/facebook_settlement/index.htm?iid=EL

[23] Y. Rachlin and D. Baron, "The secrecy of compressed sensing measurements," in *Proc. Allerton Conf. Communication, Control, and Computing* Urbana-Champaign, IL, 2008, pp. 813–817.

[24] R. L. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," in *Foundations of Secure Computation*, R. DeMillo, D. Dobkin, A. Jones, and R. Lipton, Eds. New York: Academic, 1978.

[25] A. R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *Information, Security and Cryptology—ICISC 2009* (Lecture Notes in Computer Science, vol. 5984), D. Lee and S. Hong, Eds., Berlin/Heidelberg,Germany: Springer, 2010, pp. 229–244.

[26] R. Sion and B. Carbunar, "On the computational practicality of private information retrieval," in *Proc. Network and Distributed System Security Symp.*, San Diego, CA, 2007, pp. 2006–2016.

[27] P. Smaragdis and M. Shashanka, "A framework for secure speech recognition," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 15, no. 4, pp. 1404–1413, 2007.

[28] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[29] D. Woodruff and S. Yekhanin, "A geometric approach to information theoretic private information retrieval," in *Proc. IEEE Conf. Computational Complexity*, San Jose, CA, 2005, pp. 275–284.

[30] C. Yeo, P. Ahammad, H. Zhang, and K. Ramchandran, "Rate-efficient visual correspondences using random projections," in *Proc. IEEE Int. Conf. Image Processing*, San Diego, CA, 2008, pp. 217–220.

[SP]