Efficient Fuzzy Search on Encrypted Data

Alexandra Boldyreva[1] * and Nathan Chenette[2] **

[1] Georgia Institute of Technology
sasha@gatech.edu
[2] Clemson University
nchenet@clemson.edu

Abstract. We study the problem of efficient (sub-linear) fuzzy search on encrypted outsourced data, in the symmetric-key setting. In particular, a user who stores encrypted data on a remote untrusted server forms queries that enable the server to efficiently locate the records containing the requested keywords, even though the user may misspell keywords or provide noisy data in the query. We define an appropriate primitive, for a general *closeness* function on the message space, that we call *efficiently fuzzy-searchable encryption* (*EFSE*). Next we identify an optimal security notion for EFSE. We demonstrate that existing schemes do not meet our security definition and propose a new scheme that we prove secure under basic assumptions. Unfortunately, the scheme requires large ciphertext length, but we show that, in a sense, this space-inefficiency is unavoidable for a general, optimally-secure scheme. Seeking the right balance between efficiency and security, we then show how to construct schemes that are more efficient and satisfy a weaker security notion that we propose. To illustrate, we present and analyze a more space-efficient scheme for supporting fuzzy search on biometric data that achieves the weaker notion.

1 Introduction

MOTIVATION AND RELATED WORK. Cloud storage, which is a remote storage accessed over a network, has moved from hype to reality and is currently experiencing explosive growth. One of the major challenges in cloud storage adoption is providing security against the untrusted server without compromising functionality and efficiency. Numerous works have addressed the problem of symmetric searchable encryption in recent years, e.g. [23, 12, 13, 2, 9]. The solutions differ in the level of security and efficiency they provide, however most of them only support exact-match queries.

These solutions, however, are not suitable for practical situations where queried keywords differ slightly from those corresponding to stored encrypted data. A user can use different spellings over time, such as "1 800 555-66-77" and "1(800)555 66 77". Google queries can tolerate typos, but such functionality is much more challenging to support when the data is encrypted. Moreover, data can be inherently noisy, e.g. for biometric identification: investigators querying a criminal database using data from a crime scene

* Supported by NSF: CNS-0831184 and CNS-1318511 awards.
** Part of the work done while at the Georgia Institute of Technology. Partially supported by NSF: CNS-0831184 grant of the first author.

should allow for "fuzziness" in fingerprint readings and witness description of the suspect. In this work we consider the problem of efficient (sub-linear) search on encrypted data that supports fuzzy search queries. Sub-linear and, in particular, logarithmic-time search is essential because a linear scan of the whole data is unacceptable for any application dealing with large databases. Typically, this requirement for efficient search is irreconcilable with achieving a conventional "strong" security notion. But practitioners are willing to compromise security for functionality and thus it is important to identify suitable (possibly "weak") levels of security and provide provably-secure solutions.

Several recent papers pertain to fuzzy searchable encryption. The scheme from [18] is designed to address the general problem, though it lacks formal security analysis and we later show that, in spite of being space-inefficient, its security is not strong enough. The construction from [1], as well as the related schemes for the public-key setting [10, 11] and the recent work [16] for the symmetric-key setting require the user to know all the data in advance, analyze the entire data and pre-compute the index before data outsourcing. This requirement is unsuitable for many broad applications, such as when data is frequently updated or streaming. The paper [25] motivates and discusses the problem of fuzzy search, but does not provide any solutions. Fully homomorphic encryption [14, 24] could be used to implement fuzzy search queries; however, even a (future) computationally efficient FHE scheme would require search time linear in the length of the database. Hence the task of finding a provably-secure efficient (sub-linear) fuzzy encryption scheme supporting on-the-fly encryption has been open prior to our work.

The major contribution of this work is to initiate the study of a highly relevant problem, efficient fuzzy encryption, from a cryptographic (provable-security) standpoint. It should be viewed as a "first step" in this effort and should not be considered a complete treatment of the subject, which has strong possibilities for future directions of research. Nevertheless, our work provides the foundations for the study of the subject, including basic definitions, impossibility results, and basic schemes. Our work continues a line of recent research on studying encryption schemes providing more functionality while satisfying weaker security notions, such as deterministic, order-preserving, format-preserving, property-preserving, predicate, and functional encryption [3, 9, 6, 21, 15, 17].

We now give an overview of our results.

DEFINING CLOSENESS. To even define our problem, we first need to establish what "close" means for messages. At its core, closeness is a function assigning a value ("close", "far," or "near") to any pair of messages from a space. Thus, we introduce the concept of a *closeness domain* which consists of a domain along with a closeness function.

EFFICIENTLY FUZZY-SEARCHABLE ENCRYPTION AND ITS SECURITY. Next we define the central primitive, *efficiently fuzzy-searchable encryption (EFSE)*, defined on a closeness domain. In addition to the standard functions of a symmetric encryption scheme, an EFSE scheme should provide a public function that takes a ciphertext and returns all ciphertexts in a database that are equal or close to (but none that are far from) the queried ciphertext. We also allow for optional false-positives, i.e. the function may return ciphertexts of some near messages. Furthermore, this function should be

sub-linearly efficient. We then discuss the details of how a user and the server perform search using an EFSE scheme. We note that an EFSE scheme leaks equality and "closeness" of queried messages in order to provide efficient exact-match and fuzzy search. Thus, an optimal security notion for EFSE would be a natural relaxation of the standard IND-CPA security definition prohibiting queries that trivially exploit this leakage of closeness and equality—we call this optimal security *indistinguishability under same-closeness-pattern chosen-plaintext attacks* (IND-CLS-CPA) and define it formally.

TEMPLATE EFSE CONSTRUCTION AND ITS SECURITY. For generality and convenience, we propose a general template EFSE construction providing the basis of all specific EFSE constructions that we discuss later. The template construction, which is inspired by the scheme from [18], formalizes and extends their construction by building an EFSE scheme from three elements, listed with security notions as follows.

1. An *efficient searchable encryption* (ESE) scheme, which was defined in [2] and is essentially a symmetric encryption that leaks equality, and is thus is a generalization of deterministic encryption; the relevant security notion is *indistinguishability under distinct chosen-plaintext attack* or IND-DCPA [4].
2. A *closeness-preserving tagging function* that maps domain elements to "tags" so that only close messages map to overlapping tags; the relevant security condition is called *consistency*.
3. A *batch-encoding family*, each instance of which maps batches of elements according to a deterministic function from domain to range; the relevant security notion is *privacy-preserving under chosen batch attacks* (PP-CBA) and is related to IND-DCPA.

Note that the latter two primitives and their security notions are novel.

The template scheme works as follows: a ciphertext contains an ESE-encryption of the message, as well as a batch-encoding of all of the message's "tags," as defined by the closeness-preserving tagging function. The ESE-encryption leaks equality, and the batch-encoded tags leak closeness. We show that a scheme based on the template is secure if the ESE scheme is IND-DCPA-secure, the batch-encoding family is collision-free and PP-CBA-secure, and the tagging function is consistent. We also suggest how to instantiate an IND-DCPA-secure ESE scheme and a PP-CBA-secure batch-encoding family out of blockciphers for use in constructions, leaving the remaining task of finding a consistent tagging function (discussed later, individually for each particular scheme.)

ANALYSIS OF SCHEME FROM [18]. Next, we present the first cryptographic security analysis of the scheme from [18] (which was missing a formal definition of security and proof.) We first define a scheme based on our template construction that is essentially equivalent, in that the scheme's core component is a tagging function that for a message outputs its "neighbors," i.e. the other messages in the message space that are close to a message. However, this tagging function is *not* consistent in general, which means that this construction is not IND-CLS-CPA-secure in general: to prove this, we present a simple efficient adversary with high advantage. The attack exploits a simple observation that looking at two encoded tags one can with high probability tell *how many* neighbors the associated messages share. Leaking such information is not required for the functionality of EFSE and hence is a security breach according to our definition.

We also note that the scheme from [18], besides being IND-CLS-CPA-insecure, is not very efficient in terms of ciphertext length. The constructions we propose target either strong security with the same efficiency, or much improved efficiency (with a necessarily weaker security guarantee.)

NEW OPTIMALLY-SECURE CONSTRUCTION. We propose a new general EFSE scheme. It relies on the notion of the *closeness graph*, whose vertices are the unique elements of the message space, and edges indicate closeness between elements. Defined according to the template model, the tagging function for this scheme sends a message to its set of incident edges (rather than neighboring vertices á la [18]) in the closeness graph. This tagging function is consistent, and so the scheme is IND-CLS-CPA-secure assuming the other components of the scheme satisfy the appropriate security notions.

One might worry that our construction is rather inefficient in the ciphertext length, which is linear in the maximum degree of the closeness graph. However we show that an EFSE scheme that works on general closeness domains (i.e. the scheme's algorithms do not depend on the structure of the closeness domain) must, in fact, require ciphertext length linear in the maximum degree of the closeness graph. The argument is information theoretic and relies on the functionality, rather than security, of the primitive. Thus, in achieving EFSE on arbitrarily-defined closeness domains the new IND-CLS-CPA-secure construction is (asymptotically) space-optimal, and moreover optimally secure.

CONSTRUCTIONS WITH IMPROVED EFFICIENCY. In many (even most?) practical applications, vertices of the closeness graph have massive degrees. Degrees can even be infinite, e.g. on continuous spaces—consider, for example, searching a massive database of website access-records for one that accessed a webpage at approximately 6:59:59.95 PM on May 20, 2012 (where the time query must be fuzzy to account for inherent lag-time in the network)—here, depending on the granularity of measurements and the closeness tolerance, there could be a huge number of neighbors. This situation can grow even worse for multi-dimensional spaces, as the number of "close neighbors" increases exponentially with dimensionality for closeness defined on a metric. Consider, for example, querying a criminal database with a large array of biometric measurements taken from a crime scene, in an attempt to find suspects—here, multi-dimensional closeness (closeness in every measurement) is needed, and if there are (say) a few dozen measurements, and even a narrow definition of closeness in each, the number of neighbors could again be huge. In such situations our optimally-secure scheme, as well as the less-secure scheme from [18], are unacceptably inefficient—and the aforementioned lower bound result shows that we cannot expect to do better for arbitrary domains.

We seek the right balance between the desired efficiency and security of EFSE, and look at closeness domains with a well-defined structure. We argue that IND-CLS-CPA-security is too strong to be useful in characterizing EFSEs on "non-rigid" closeness domains (where near messages could be encrypted to either close or far ciphertexts), and so to do this we introduce a new security definition. The new definition requires schemes to hide all information about plaintexts except nearness and a certain aspect of "local structure"—essentially, messages' offsets from a predetermined fixed regular lattice $\mathcal{L}$ on the space. Importantly, this implies that no major relative information (i.e., nothing above the least-significant-bit level) is leaked about a pair of "disconnected messages," that is, messages that cannot be connected through a chain of near known correspond-

ing ciphertext pairs. Hence, we call this notion *macrostructure-security*. Note that this security may be useful in applications such as the website access-record and biometric matching examples above, where it is not a big deal to reveal aspects of local structure (does it matter if an adversary knows, say, the least significant bits relating to biometric measurements in the criminal database?) but it is important to hide large differences between messages.

Our security definition and construction strategy focus on a practical choice of domains with associated metric and close, near, and far distance thresholds, that we call *metric closeness domains*; in particular, we consider real multidimensional space. Critically, on these domains, closeness is defined in a "regular" manner across the space—namely, for any regular lattice in the space, closeness is invariant under translation by a lattice vector. The security definition is then defined in terms of a fixed lattice, demanding that nothing is leaked except "local structure" of near clusters of messages with respect to the lattice. To provide a blueprint for building specific schemes, we introduce the concept of an "anchor radius" for a metric closeness domain and a lattice, and use it to construct a tagging function to build an EFSE via our usual template. We show that a valid anchor radius implies an EFSE construction that is macrostructure-secure. Then, to enhance understanding, we present a practical example, filling in details of the blueprint to build a (relatively) space-efficient, macrostructure-secure EFSE scheme supporting fuzzy search on fingerprint data. Finally, in the full version [8], we observe that an efficient scheme that probabilistically acts like an EFSE scheme can be constructed out of locality-sensitive hash (LSH) functions. But the theory behind these schemes and their security is beyond the scope of this work.

FUTURE WORK. Our work provides the basis for cryptographic study of fuzzy-searchable encryption. Our template constructions invite exploration of more efficient schemes that will automatically satisfy our security notions. In addition, future studies might achieve more efficient and secure schemes—circumventing our impossibility result by defining closeness and EFSE primitives in a different manner. For instance, one could consider only closeness domains with certain natural structure, or closeness could be defined quantitatively or probabilistically.

2 Preliminaries

We let $\mathcal{LR}$ (left-or-right) denote the "selector" that on input m_0, m_1, b returns m_b. For $x \in \mathbb{Z}$, the notation $[x]$ denotes the set $\{1, 2, \ldots, x\}$. In some of the algorithm descriptions, for ease and clarity of analysis, we use abstract set notation. In a practical implementation, the sets can be implemented by some specialized data structure, or by vectors/lists with a common predetermined order (e.g., numerical order.) We recall the syntax and security for symmetric encryption in the full version of the paper [8]. We wait until Section 4 to define efficiently searchable encryption, privacy-preserving batch-encoding, and closeness-preserving tagging functions. Here, we introduce a metric space, closeness domains and associated graph-theoretical concepts.

METRIC SPACES. $(\mathcal{D}, d)$ is a *metric space* if $\mathcal{D}$ is a set and d (the *metric*) is a real-valued function on $\mathcal{D} \times \mathcal{D}$ such that for all $x, y, z \in \mathcal{D}$,

$$d(x,y) \geq 0 \qquad d(x,y) = 0 \text{ iff } x = y$$
$$d(x,y) = d(y,x) \qquad d(x,z) \leq d(x,y) + d(y,z).$$

CLOSENESS DOMAIN. We refer to the pair $\Lambda = (\mathcal{D}, \mathrm{Cl})$ as a *closeness domain* if

1. $\mathcal{D}$ is a (finite or infinite) set, called the *domain* or *message space*;
2. Cl is the *closeness function* that takes a pair of messages and outputs a member of $\{\mathtt{eq}, \mathtt{close}, \mathtt{near}, \mathtt{far}\}$, so that Cl is symmetric (i.e., $\mathrm{Cl}(m, m') = \mathrm{Cl}(m', m)$ for all $m, m' \in \mathcal{D}$) and $\mathrm{Cl}(m, m') = \mathtt{eq}$ if and only if $m = m'$.

According to the output of Cl, we say a pair of messages is *equal*, *close*, *near*, or *far*. Note that a closeness domain can be defined by describing which distinct message pairs of a domain $\mathcal{D}$ are close and which are far (the rest are then near.) For convenience, we say Λ is *rigid* if $\mathrm{Cl}(m, m') \in \{\mathtt{close}, \mathtt{far}\}$ for all $m \neq m' \in \mathcal{D}$. When these quantities exist, the *degree* of a message m in Λ is $\Delta_m = |\{m' \in \mathcal{D} \mid \mathrm{Cl}(m, m') = \mathtt{close}\}|$, and the *max degree* of Λ is $\Delta = max_{m \in \mathcal{D}} \Delta_m$.

As a special case, let d be a metric[3] on domain $\mathcal{D}$, and let $\delta > 0$. The *metric* closeness domain $\left(\mathcal{D}, \mathcal{M}_d^{\delta^{\mathtt{C}}, \delta^{\mathtt{F}}}\right)$ on domain $\mathcal{D}$ with respect to metric d, *close threshold* $\delta^{\mathtt{C}} \geq 0$, and *far threshold* $\delta^{\mathtt{F}} \geq \delta^{\mathtt{C}}$, has the following closeness function: for distinct $m, m' \in \mathcal{D}$, $\mathcal{M}_d^{\delta^{\mathtt{C}}, \delta^{\mathtt{F}}} = \begin{cases} \mathtt{close} & \text{if } d(m,m') \leq \delta^{\mathtt{C}}; \\ \mathtt{far} & \text{if } d(m,m') > \delta^{\mathtt{F}}. \end{cases}$ For instance, $\left(\{0,1\}^{80}, \mathcal{M}_{\mathrm{Ham}}^{1,2}\right)$, where Ham is Hamming distance, is a closeness domain of all length-80 strings where strings differing in 1 bit are close, differing in 2 bits are near, and differing in more than 2 bits are far.

CLOSENESS AND NEARNESS GRAPH, INDUCED SUBGRAPH. Let $\Lambda = (\mathcal{D}, \mathrm{Cl})$ be a closeness domain, $\mathcal{V}_\Lambda = \mathcal{D}$ and

$$\mathcal{E}_\Lambda^{\mathtt{C}} = \{\{u, v\} \mid u \neq v \in \mathcal{V}_\Lambda \text{ and } \mathrm{Cl}(u, v) = \mathtt{close}\};$$
$$\mathcal{E}_\Lambda^{\mathtt{N}} = \{\{u, v\} \mid u \neq v \in \mathcal{V}_\Lambda \text{ and } \mathrm{Cl}(u, v) \in \{\mathtt{close}, \mathtt{near}\}\}.$$

Then $\mathcal{G}_\Lambda^{\mathtt{C}} = (\mathcal{D}, \mathcal{E}_\Lambda^{\mathtt{C}})$ is the *closeness graph* and $\mathcal{G}_\Lambda^{\mathtt{N}} = (\mathcal{D}, \mathcal{E}_\Lambda^{\mathtt{N}})$ is the *nearness graph* of Λ. For graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $H \subseteq \mathcal{V}$ let $\mathcal{G}(H) = (H, \mathcal{E}(H))$ be the *subgraph induced by* H where $\mathcal{E}(H) = \{\{u, v\} \in \mathcal{E} \mid u, v \in H\}$.

3 Efficiently Fuzzy-Searchable Symmetric Encryption

We now define our main primitive and show how can it be used for efficient search. Following that, we formulate the optimal level of security for EFSE schemes.

DEFINING EFFICIENTLY FUZZY-SEARCHABLE ENCRYPTION. $\mathtt{FSE} = (\mathcal{K}, \mathcal{E}nc, \mathcal{D}ec, \mathtt{makeDS}, \mathtt{fuzzyQ})$ is a *structured fuzzy-searchable symmetric encryption* (StructFSE)

[3] So in particular, d obeys the triangle inequality.

scheme on closeness domain $\Lambda = (\mathcal{D}, \mathrm{Cl})$ if $(\mathcal{K}, \mathcal{E}nc, \mathcal{D}ec)$ is a symmetric encryption scheme on $\mathcal{D}$, and for any key K output by $\mathcal{K}$,

- $\mathtt{makeDS}$ takes a set of ciphertexts $\mathbf{C}$ (the *database*) encrypted under K and outputs a data structure $\mathtt{DS}_{\mathbf{C}}$;
- $\mathtt{fuzzyQ}$, given database $\mathbf{C}$, data structure $\mathtt{DS}_{\mathbf{C}}$, and query ciphertext c, outputs two subsets $\mathbf{E}, \mathbf{F}$ of $\mathbf{C}$ such that

$$\mathbf{E} = \mathbf{C}_{\mathtt{eq}}(c) \qquad \text{and} \qquad \mathbf{C}_{\mathtt{close}}(c) \subseteq \mathbf{F} \subseteq \mathbf{C}_{\mathtt{near}}(c),$$

where for $m = \mathcal{D}ec(K, c)$, $m' = \mathcal{D}ec(K, c')$,

$$\begin{cases} \mathbf{C}_{\mathtt{eq}}(c) & = \{c' \in \mathbf{C} \mid \mathrm{Cl}(m, m') = \mathtt{eq}\} \\ \mathbf{C}_{\mathtt{close}}(c) & = \{c' \in \mathbf{C} \mid \mathrm{Cl}(m, m') = \mathtt{close}\}. \\ \mathbf{C}_{\mathtt{near}}(c) & = \{c' \in \mathbf{C} \mid \mathrm{Cl}(m, m') \in \{\mathtt{close}, \mathtt{near}\}\}. \end{cases}$$

One could easily relax the above syntax to not require the returned ciphertexts to equal those from the database. This would allow one to consider, for example, schemes based on homomorphic encryption. We stick with a stricter definition for simplicity. To ease discussion, for implicit fixed key K we say that ciphertexts c and c' are close (respectively, far) if their decryptions $m = \mathcal{D}ec(K, c)$ and $m' = \mathcal{D}ec(K, c')$ are close (far). Notice that in a StructFSE scheme, $\mathtt{fuzzyQ}(\mathbf{C}, \mathtt{DS}_{\mathbf{C}}, c)$ returns all ciphertexts in $\mathbf{C}$ close to c and no ciphertexts far from c. Any near ciphertext may be returned as well—these can be thought of as "legal false positives" in a fuzzy search query. In this sense, FSE on a rigid closeness domain cannot have any false positives. But of course, even on a non-rigid domain, we must limit false positives to ensure efficiency.

We say StructFSE scheme $\mathtt{FSE} = (\mathcal{K}, \mathcal{E}nc, \mathcal{D}ec, \mathtt{makeDS}, \mathtt{fuzzyQ})$ is an *efficiently fuzzy searchable symmetric encryption* (EFSE) scheme if for any (sufficiently large) database $\mathbf{C}$, data structure $\mathtt{DS}_{\mathbf{C}}$, key K generated by $\mathcal{K}$, and query ciphertext c with $|\mathbf{C}_{\mathtt{close}}(c)|$ sub-linear in the size of $\mathbf{C}$, the running time of $\mathtt{fuzzyQ}_{\mathbf{C},\mathtt{DS}_{\mathbf{C}}}(c)$ is sub-linear in the size of $\mathbf{C}$. Notice this condition on the running time limits the number of false positives for a fuzzy query.

We note that EFSE defined for rigid domains makes a special case of property-preserving encryption from [21] (for the property of "closeness"), but the general case of EFSE does not seem to fit the class of schemes from [21].

USING AN EFSE SCHEME. Let $\mathtt{FSE} = (\mathcal{K}, \mathcal{E}nc, \mathcal{D}ec, \mathtt{makeDS}, \mathtt{fuzzyQ})$ be an EFSE scheme and K a valid key. In a practical scenario, let $\mathbf{C}$ be the set of ciphertexts currently in an encrypted database, encrypted under K. The server runs $\mathtt{makeDS}(\mathbf{C})$ to create a data structure $\mathtt{DS}_{\mathbf{C}}$, and upon a new query $c = \mathcal{E}nc_K(m)$, runs $\mathtt{fuzzyQ}(\mathbf{C}, \mathtt{DS}_{\mathbf{C}}, c)$ and returns the results, $\mathbf{E}$ and $\mathbf{F}$, to the user. By correctness of the scheme, $\mathbf{F}$ consists of all ciphertexts in $\mathbf{C}$ whose messages are close to m, and no ciphertexts whose messages are far from m. Since the scheme is efficient, such a query will take time sub-linear in the size of the database $\mathbf{C}$ (assuming the number of close messages itself is also sub-linear in the size of $\mathbf{C}$.) Also note that the scheme supports efficient exact-match search through $\mathbf{E}$.

As a side note, in a practical implementation, additional functions (e.g. add, remove, edit) would be useful to efficiently update the data structure as the database changes.

In our analysis, we are less focused on efficiency of the data structure maintenance, so for simplicity we just let the (possibly inefficient) function `makeDS` construct the data structure from the entire database. And we leave it as an interesting open problem for future work to extend and realize the primitive so that "closeness" be specified during encryption.

Finally, observe that the "difficult" part of building an EFSE scheme is ensuring that `fuzzyQ` is efficient. Thus, the construction of $\mathcal{E}nc$ might as well be designed with the efficiency of `fuzzyQ` in mind. In our constructions, as detailed in Section 4, ciphertexts outputted by $\mathcal{E}nc$ will contain "encoded tags" such that ciphertexts of close messages share a common encoded tag. Thus, indexing ciphertexts by encoded tags in an efficiently searchable data structure, like a binary search tree, leads to an efficient construction of `fuzzyQ`.

OPTIMAL SECURITY FOR EFSE SCHEMES. We construct the following indistinguishability-based security definition, called IND-CLS-CPA[4], for analyzing the security of EFSE schemes. Intuitively, this notion is identical to IND-CPA with the additional condition that left-right queries have the same *closeness pattern* (in the second requirement below.)

Definition 1. *Let* `FSE` *be an EFSE scheme on closeness domain* $\Lambda = (\mathcal{D}, \mathrm{Cl})$. *For bit* $b \in \{0,1\}$ *and adversary* A, *let* $\mathbf{Exp}^{\text{ind-cls-cpa-}b}_{\mathtt{FSE}}(A)$ *be the standard IND-CPA experiment* $\mathbf{Exp}^{\text{ind-cpa-}b}_{\mathtt{FSE}}(A)$ *recalled in Figure 1, but with the following restriction: if* $(m_0^1, m_1^1), \ldots, (m_0^q, m_1^q)$ *are the queries* A *makes to its LR encryption oracle* $\mathcal{E}nc(K, \mathcal{LR}(\cdot,\cdot,b))$, *then*

1. $|m_0^i| = |m_1^i|$ *for all* $i \in [q]$;
2. *for all* $i, j \in [q]$, $\mathrm{Cl}(m_0^i, m_0^j) = \mathrm{Cl}(m_1^i, m_1^j)$.

For an adversary A, *define its* IND-CLS-CPA advantage *against* `FSE` *as*

$$\mathbf{Adv}^{\text{ind-cls-cpa}}_{\mathtt{FSE}}(A) \;=\; \Pr\left[\mathbf{Exp}^{\text{ind-cls-cpa-1}}_{\mathtt{FSE}}(A) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-cls-cpa-0}}_{\mathtt{FSE}}(A) = 1\right].$$

We say that `FSE` *is* indistinguishable under same-closeness-pattern chosen-plaintext attacks *(IND-CLS-CPA-secure) if the IND-CLS-CPA advantage of any adversary against* `FSE` *is small*[5][6].

[4] We do not study chosen-ciphertext security here as it can be achieved using the encrypt-then-MAC method [5].

[5] We use the informal term "small" because the main building blocks of symmetric cryptography, blockciphers, have keys of fixed length in practice. Thus, instead of requiring advantages to be negligible in a security parameter, we leave appropriate concrete bounds to be determined on a case-by-case basis depending on the application.

[6] According to our definitions, advantage can be negative; note that "small" refers to an advantage close to zero. For every adversary with negative advantage there is one with positive advantage, who just outputs the complement bit.

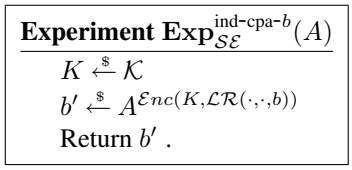

Fig. 1. The IND-CPA experiment.

It should be apparent that IND-CLS-CPA-security is optimal for EFSE schemes on *rigid* closeness domains: revealing equality/closeness patterns of LR-queries is unavoidable as an adversary can run the (public) `fuzzyQ` function on ciphertexts to test for equality and closeness. It may seem that the optimal security definition on *general* closeness domains, where `fuzzyQ` is given flexibility over near message pairs, should not allow distinguishing near messages as it is not needed for search functionality. However, while a stronger security definition than IND-CLS-CPA would be possible, the notion would necessarily depend on the scheme's construction, i.e., the left-right query restriction would rely on how `fuzzyQ` sends near message pairs to close or far ciphertexts. To define a security notion that is independent from the construction of `fuzzyQ`, the IND-CLS-CPA experiment forces left-right query pairs to match near-to-near, as `fuzzyQ` is permitted to distinguish near ciphertexts from close and far ciphertexts.

4 Template tag-encoding construction for EFSE

In this somewhat technical section, we build up to a general construction of an EFSE scheme given a valid "tagging function" on the desired closeness domain. In addition, we show that under certain conditions, the scheme is IND-CLS-CPA-secure. First, though, we define several primitives, along with relevant security notions, that will be components of the construction. The primitives are: efficient searchable encryption (ESE) schemes [2], closeness-preserving tagging functions, and privacy-preserving batch-encoding families. We emphasize that, despite the technical language, these primitives are conceptually simple and can be instantiated in natural ways—the formalism is simply aimed to achieve fuller generality in isolating theoretical requirements from possible instantiations.

EFFICIENT SEARCHABLE ENCRYPTION AND SECURITY. The ESE scheme primitive [2] is recalled in the full version [8]. Intuitively, an ESE is an encryption scheme that "leaks equality," that is, there is a (public) way to tell if two ciphertexts are encryptions of the same message. In particular, deterministic functions F, G are provided such that if c_1 and c_2 are both encryptions of m under key K, $G(c_1) = F(K, m) = G(c_2)$ (and this is unlikely if c_1 and c_2 are encryptions of different messages.) The appropriate security notion for ESE was defined by [4] and is called *indistinguishability under distinct chosen plaintext attacks* (IND-DCPA)—it is also recalled in the full version [8]. The notion is identical to IND-CPA except that LR-queries must have the same "equality pattern" (and so avoiding the obvious attack, as ESE leaks equality.) Note that any PRF implies an IND-DCPA-secure ESE scheme [4] so there are many options for instantiation.

CLOSENESS-PRESERVING TAGGING FUNCTIONS. Fix a closeness domain $\Lambda = (\mathcal{D}, \text{Cl})$. Let `TagUniv` be a (finite or infinite) set and let $\texttt{Tags} : \mathcal{D} \to 2^{\texttt{TagUniv}}$ be a function assigning a subset of `TagUniv` to every domain element. We call `Tags` a *closeness-preserving tagging function* (CPTF) from Λ into `TagUniv` if for every $x, y \in \mathcal{D}$ with

$\mathrm{Cl}(x, y) = \texttt{close}$, there exists $t \in \texttt{TagUniv}$ such that $t \in \texttt{Tags}(x) \cap \texttt{Tags}(y)$; and for every $x, y \in \mathcal{D}$ with $\mathrm{Cl}(x, y) = \texttt{far}$, $\texttt{Tags}(x) \cap \texttt{Tags}(y) = \emptyset$.

Further, a CPTF Tags is *consistent* with respect to closeness domain Λ if for any message sets $\{m_0^1, \ldots, m_0^q\}$ and $\{m_1^1, \ldots, m_1^q\}$ having the same closeness pattern[7], we have $\left|\bigcap_{i\in[q]} \texttt{Tags}(m_0^i)\right| = \left|\bigcap_{i\in[q]} \texttt{Tags}(m_1^i)\right|$. Consistency can be understood intuitively as follows: whenever a set of messages has the same closeness pattern as another set of messages, each set should have the same number of common tags.

Examples of CPTFs are integral to our constructions and several are introduced in the remainder of this paper.

PRIVACY-PRESERVING BATCH-ENCODING. We say that $\mathcal{F} = (\mathcal{K}, \texttt{En})$ is an *encoding family* on domain $\mathcal{D}$ and range $\mathcal{R}$ if $\mathcal{K}$ outputs random keys and En takes a key K and an element of $\mathcal{D}$ and outputs an element of $\mathcal{R}$ such that $\texttt{En}(K, \cdot)$ is a (deterministic) function from $\mathcal{D}$ to $\mathcal{R}$. We further say that $\mathcal{F}_{\texttt{Ben}} = (\mathcal{K}_{\texttt{Ben}}, \texttt{En}, \texttt{Ben})$ is a *batch-encoding family* if $(\mathcal{K}_{\texttt{Ben}}, \texttt{En})$ is an encoding family and Ben takes a key K and a set of elements $M \subseteq \mathcal{D}$ and outputs $\{\texttt{En}(K, m) \mid m \in M\}$. Given a function family $(\mathcal{K}', \texttt{En}')$ it is easy to construct a batch-encoding family $(\mathcal{K}_{\texttt{Ben}}, \texttt{En}, \texttt{Ben})$: let $\mathcal{K}_{\texttt{Ben}} = \mathcal{K}'$ and $\texttt{En} = \texttt{En}'$, and define $\texttt{Ben}(K, \cdot)$ to take a set of messages, run $\texttt{En}(K, \cdot)$ on each, and return the set of results.

Experiment $\mathbf{Exp}^{\text{pp-cba-}b}_{\mathcal{F}_{\texttt{Ben}}}(A)$

$K \xleftarrow{\$} \mathcal{K}_{\texttt{Ben}}$

$b' \xleftarrow{\$} A^{\texttt{Ben}(K, \mathcal{LR}(\cdot,\cdot,b))}$

Return b' ,

Fig. 2. The PP-CBA experiment.

We say that a encoding family $(\mathcal{K}_{\texttt{Ben}}, \texttt{En})$ or a batch-encoding family $(\mathcal{K}_{\texttt{Ben}}, \texttt{En}, \texttt{Ben})$ is *collision-free* if for any key K, $\texttt{En}(K, \cdot)$ is one-to-one on $\mathcal{D}$. Now, we define security for batch-encoding families. Called *privacy-preserving under chosen batch attacks*, it is essentially the IND-DCPA generalized to objects of the batch-encoding primitive.

Definition 2. *Let $\mathcal{F}_{\texttt{Ben}} = (\mathcal{K}_{\texttt{Ben}}, \texttt{En}, \texttt{Ben})$ be a batch-encoding family on domain $\mathcal{D}$ and range $\mathcal{R}$. For an adversary A and $b \in \{0, 1\}$ consider the experiment defined in Figure 2, where it is required that, if $(M_0^1, M_1^1), \ldots, (M_0^q, M_1^q)$ are the queries that A makes to its $\mathcal{LR}$-batch-encoding oracle (note: each M_j^i is a* set *of elements of $\mathcal{D}$), for all $I \subseteq [q]$ we have $\left|\bigcap_{i\in I} M_0^i\right| = \left|\bigcap_{i\in I} M_1^i\right|$.*

For an adversary A, define its PP-CBA advantage *against $\mathcal{F}_{\texttt{Ben}}$ as*

$$\mathbf{Adv}^{\text{pp-cba}}_{\mathcal{F}_{\texttt{Ben}}}(A) \;=\; \Pr\left[\,\mathbf{Exp}^{\text{pp-cba-1}}_{\mathcal{F}_{\texttt{Ben}}}(A) = 1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{pp-cba-0}}_{\mathcal{F}_{\texttt{Ben}}}(A) = 1\,\right] .$$

We say that $\mathcal{F}_{\texttt{Ben}}$ is privacy-preserving under chosen batch attacks *(PP-CBA-secure) if the PP-CBA advantage of any adversary against $\mathcal{F}_{\texttt{Ben}}$ is small.*

Notice that the requirement rules out an obvious attack: suppose to the contrary that, without loss of generality, the adversary could query $(M_0^1, M_1^1), \ldots, (M_0^q, M_1^q)$ with $\left|\bigcap_{i\in[q]} M_0^i\right| > \left|\bigcap_{i\in[q]} M_1^i\right|$. If $\texttt{En}(K, \cdot)$ is collision-free, $\left|\bigcap_{i\in[q]} \texttt{Ben}(K, M_b^i)\right| =$

[7] That is, $\mathrm{Cl}(m_0^i, m_0^j) = \mathrm{Cl}(m_1^i, m_1^j)$ for all $i, j \in [q]$.

$\left|\bigcap_{i\in[q]}\{\mathtt{En}(K,m) \mid m \in M_b^i\}\right| = \left|\bigcap_{i\in[q]} M_b^i\right|$, so by computing $\left|\bigcap_{i\in[q]} \mathtt{Ben}(K, M_b^i)\right|$ from the oracle responses the adversary can identify b.

ON HOW TO INSTANTIATE A PRIVACY-PRESERVING, COLLISION-FREE BATCH-ENCODING SCHEME. Anticipating that our EFSE constructs will use PP-CBA-secure batch-encoding schemes, how can we construct one? In fact, a PP-CBA-secure batch-encoding scheme can be created straightforwardly out of a pseudorandom function (PRF), as we now demonstrate.

Let $\mathtt{PRF} = (\mathcal{K}_{\mathtt{PRF}}, \mathcal{F}_{\mathtt{PRF}})$ be a function family on domain $\mathcal{D}$ to some range $\mathcal{R}$. Let $\mathcal{F}_{\mathtt{Ben}} = (\mathcal{K}_{\mathtt{Ben}}, \mathtt{En}, \mathtt{Ben})$ where $\mathcal{K}_{\mathtt{Ben}} = \mathcal{K}_{\mathtt{PRF}}$, $\mathtt{En} = \mathcal{F}_{\mathtt{PRF}}$, and $\mathtt{Ben}$ is defined in the standard way using $\mathtt{En}$ as described above. We claim that if $\mathtt{PRF}$ is a PRF, then $\mathcal{F}_{\mathtt{Ben}}$ is PP-CBA-secure. See the following result, which is proved in [8].

Proposition 1. *For $\mathcal{F}_{\mathtt{Ben}}$ constructed as above out of function family $\mathtt{PRF}$, and any adversary A, there exist PRF adversaries F_0 and F_1 such that*

$$\mathbf{Adv}^{\text{pp-cba}}_{\mathcal{F}_{\mathtt{Ben}}}(A) = \mathbf{Adv}^{\text{prf}}_{\mathtt{PRF}}(F_0) + \mathbf{Adv}^{\text{prf}}_{\mathtt{PRF}}(F_1)\ .$$

Further, if A submits queries of total length γ to its oracle, then F_1 and F_2 each submit queries of total length γ to their oracles as well.

As will soon become clear, what we actually need is a PP-CBA-secure *collision-free* batch-encoding scheme, a natural extension of a IND-DCPA deterministic encryption scheme. To theoretically achieve collision resistance, a pseudorandom permutation (PRP) would be necessary. But concretely, statistical collision resistance should suffice—i.e. on random inputs, a collision occurs after $\sqrt{|\mathcal{R}|}$ inputs with probability approximately $1/2$. We suggest using any blockcipher (permutation) that is a PRF (and thus PP-CBA-secure), though one may have to augment the blockcipher into a variable-input-length blockcipher [7] as described in [20], or into an encryption scheme like those of [22, 2].

TEMPLATE TAG-ENCODING EFSE CONSTRUCTION. We now provide a general "template" construction for an EFSE scheme given a closeness-preserving tagging function $\mathtt{Tags}$, batch-encoding family $\mathcal{F}_{\mathtt{Ben}}$, and ESE scheme $\mathtt{ESE}$. We remark that this template is a generalization of the technique used in [18], though we have expanded, formalized, and refined it significantly. All forthcoming EFSE constructions use this general construction as a template.

Let $\Lambda = (\mathcal{D}, \mathrm{Cl})$ be a closeness domain, $\mathtt{Tags}$ a function from $\mathcal{D}$ to subsets of a set $\mathtt{TagUniv}$, $\mathcal{F}_{\mathtt{Ben}} = (\mathcal{K}_{\mathtt{Ben}}, \mathtt{En}, \mathtt{Ben})$ a batch-encoding family on domain $\mathcal{D}_{\mathtt{En}} = \mathtt{TagUniv}$ and range $\mathcal{R}_{\mathtt{En}}$, and $\mathtt{ESE} = (\mathcal{K}_{\mathtt{ESE}}, \mathcal{E}nc_{\mathtt{ESE}}, \mathcal{D}ec_{\mathtt{ESE}}, F, G)$ an ESE scheme on $\mathcal{D}$. Then we define a general *tag-encoding* StructFSE scheme $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ in Figure 3.

CONDITIONS FOR CORRECTNESS AND EFFICIENCY. The following result, proved in [8], establishes conditions under which the template construction is a valid StructFSE scheme and when it is EFSE.

Theorem 1. *If $\mathcal{F}_{\mathtt{Ben}}$ is collision-free and $\mathtt{Tags}$ is closeness-preserving, then the scheme $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ is StructFSE. In addition, it is an EFSE scheme if $\mathtt{Tags}$, $\mathcal{F}_{\mathtt{Ben}}$, and $\mathtt{ESE}$ are efficient and $\mu = \max_m |\mathtt{Tags}(m)|$ is small.*

$\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}] = (\mathcal{K}, \mathcal{E}nc, \mathcal{D}ec, \mathtt{makeDS}, \mathtt{fuzzyQ})$ where

- $\mathcal{K}$ runs $K_{\mathtt{Ben}} \xleftarrow{\$} \mathcal{K}_{\mathtt{Ben}}$ and $K_{\mathtt{ESE}} \xleftarrow{\$} \mathcal{K}_{\mathtt{ESE}}$, and returns $K_{\mathtt{Ben}} \| K_{\mathtt{ESE}}$.
- $\mathcal{E}nc(K_{\mathtt{Ben}} \| K_{\mathtt{ESE}}, m)$ runs $T_m \leftarrow \mathtt{Tags}(m)$; $\mathtt{Etags} \leftarrow \mathtt{Ben}(K_{\mathtt{Ben}}, T_m)$; $c_R \leftarrow \mathcal{E}nc_{\mathtt{ESE}}(K_{\mathtt{ESE}}, m)$, and returns $c \leftarrow \mathtt{Etags} \| c_R$.
- $\mathcal{D}ec(K_{\mathtt{Ben}} \| K_{\mathtt{ESE}}, c)$ parses c as $\mathtt{Etags} \| c_R$ and returns $\mathcal{D}ec_{\mathtt{ESE}}(K_{\mathtt{ESE}}, c_R)$.
- $\mathtt{makeDS}(\mathbf{C})$ initializes an efficient self-balancing search tree $\mathcal{T}$ representing an associative array from elements of $\mathcal{R}_{\mathtt{En}}$ to ciphertexts. For each ciphertext $c \in \mathbf{C}$ parsed as $c = \mathtt{Etags} \| c_R$, and for each $t \in \mathtt{Etags}$, add the node $(t \mapsto c)$ to $\mathcal{T}$. Output $\mathtt{DS}_{\mathbf{C}} \leftarrow \mathcal{T}$.
- $\mathtt{fuzzyQ}_{\mathbf{C}, \mathtt{DS}_{\mathbf{C}}}(c)$ parses c as $\mathtt{Etags} \| c_R$ and interprets $\mathtt{DS}_{\mathbf{C}}$ as search tree $\mathcal{T}$. Let $\mathbf{E}, \mathbf{F} = \emptyset$. For each $t \in \mathtt{Etags}$, search $\mathcal{T}$ for nodes indexed by t; for any $(t \mapsto c')$ that exist, parse $c' = \mathtt{Etags}' \| c'_R$. Then, if $G(c_R) = G(c'_R)$, add c' to $\mathbf{E}$; otherwise, add c' to $\mathbf{F}$. Return $\mathbf{E}, \mathbf{F}$.

Fig. 3. General tag-encoding construction of a StructFSE scheme given $\mathtt{Tags}$, $\mathcal{F}_{\mathtt{Ben}}$, $\mathtt{ESE}$.

CONDITIONS FOR OPTIMAL SECURITY. Now, fix a closeness domain $\Lambda = (\mathcal{D}, \mathrm{Cl})$, and let $\mathtt{Tags}$ be a CPTF from Λ into a set $\mathtt{TagUniv}$, $\mathcal{F}_{\mathtt{Ben}}$ a collision-free batch-encoding family on $\mathtt{TagUniv}$, and $\mathtt{ESE}$ an ESE scheme on $\mathcal{D}$, so that $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ is a valid StructFSE scheme by Theorem 1. The next result, proved in [8], gives sufficient conditions for $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ to be IND-CLS-CPA-secure.

Theorem 2. *If* $\mathtt{Tags}$ *is consistent with respect to* Λ*,* $\mu = \max_m |\mathtt{Tags}(m)|$ *is small,* $\mathcal{F}_{\mathtt{Ben}}$ *is PP-CBA-secure, and* $\mathtt{ESE}$ *is IND-DCPA-secure, then* $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ *is IND-CLS-CPA-secure.*

Finally, the following result, proved in [8], shows that consistency of $\mathtt{Tags}$ is a necessary condition for the template scheme to be IND-CLS-CPA-secure.

Theorem 3. *If* $\mathtt{Tags}$ *is not consistent, then valid EFSE* $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ *is not IND-CLS-CPA-secure.*

Summing up, if CPTF $\mathtt{Tags}$ is consistent, $\mu = \max_m |\mathtt{Tags}(m)|$ is small, batch-encoding oracle $\mathcal{F}_{\mathtt{Ben}}$ is PP-CBA-secure *and* collision-free, and ESE scheme $\mathtt{ESE}$ is IND-DCPA-secure, then $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{Tags}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ is a valid, (optimally) IND-CLS-CPA-secure EFSE. If $\mathtt{Tags}$ is not consistent, the scheme is not IND-CLS-CPA-secure.

5 Toward an Optimally-Secure Scheme

We now seek an EFSE construction achieving the optimal level of security, IND-CLS-CPA, as defined in Definition 1. First, we show that the only previously existing candidate is, in general, not IND-CLS-CPA-secure due to Theorem 3. Then, we construct the first IND-CLS-CPA-secure EFSE scheme using the template from Section 4. Finally, we show that in a sense, the space-inefficiency of the secure scheme is necessary to accommodate general closeness domains.

ANALYSIS OF AN EFSE SCHEME SIMILAR TO [18]. The only previously existing EFSE-type scheme is presented in [18]. As noted, the basic structure of our template tag-encoding scheme is a generalization of their method, so it is natural to define a tag-encoding scheme in our model that captures the essence of (and perhaps improves) the [18] scheme. Here we show that this scheme has poor space-efficiency (length of ciphertext linear in the degree of a message) and yet fails to achieve IND-CLS-CPA-security. (Moreover, it only works on certain closeness domains.) In contrast, the schemes we develop in later sections either achieve IND-CLS-CPA-security, or have much better space-efficiency.

In [18], the authors construct several variants of a fuzzy-searchable scheme; here we present a variant/generalization[8]. This construction only works on closeness domains $\Lambda = (\mathcal{D}, \mathrm{Cl})$ with the following constraint: for any $m_1, m_2 \in \mathcal{D}$, if $\mathrm{Cl}(m_1, m_2) = \texttt{far}$, then there exists no m with $\mathrm{Cl}(m_1, m) = \mathrm{Cl}(m_2, m) = \texttt{close}$. (In particular, this generally rules out rigid closeness domains.) We define the *neighbor set* of an element m to be $\texttt{Nb}_m = \{m' \in \mathcal{D} \mid m' \neq m, \mathrm{Cl}(m, m') = \texttt{close}\}$. Define $\texttt{TagNbs} : \mathcal{D} \to \mathcal{V}_\Lambda$ as $\texttt{TagNbs}(m) = \texttt{Nb}_m \cup \{m\}$, where $\mathcal{V}_\Lambda$ is the power set of $\mathcal{D}$.

Note that if $\mathrm{Cl}(m, m') = \texttt{close}$ then $\texttt{TagNbs}(m) \cap \texttt{TagNbs}(m') \supseteq \{m, m'\} \neq \emptyset$, and if $\mathrm{Cl}(m, m') = \texttt{far}$, $\texttt{TagNbs}(m) \cap \texttt{TagNbs}(m') = \emptyset$ by the condition on Λ, so TagNbs is a CPTF on Λ. Let $\mathcal{F}_{\texttt{Ben}}$ be a collision-free batch-encoding family on $\mathcal{V}_\Lambda$ and ESE an ESE scheme on $\mathcal{D}$, and define FSEtagNbs to be $\texttt{FSE}_{\texttt{Etag}}[\texttt{TagNbs}, \mathcal{F}_{\texttt{Ben}}, \texttt{ESE}]$ as per Figure 3. If the max degree $\Delta = \max_{m \in \mathcal{D}} |\texttt{Nb}_m|$ of Λ is small, FSEtagNbs is an EFSE. However, the ciphertext size is linear in Δ.

We claim that FSEtagNbs is IND-CLS-CPA-insecure for the closeness domains considered by [18], as well as most other conceivably useful domains. Suppose, for example, that the closeness domain has two pairs of close messages with different numbers of common close neighbors: i.e.,

$$\mathrm{Cl}(m_0, m_2) = \mathrm{Cl}(m_1, m_2) = \texttt{close}\,; \qquad |\texttt{Nb}_{m_0} \cap \texttt{Nb}_{m_2}| \neq |\texttt{Nb}_{m_1} \cap \texttt{Nb}_{m_2}|. \qquad (1)$$

Then the condition of Theorem 3 is satisfied for $q = 2$, so that FSEtagNbs is IND-CLS-CPA-insecure for any domain having m_0, m_1, m_2 that satisfy (1).

The schemes of [18] are, essentially, instantiations of FSEtagNbs on closeness domains defined in terms of keywords and edit distance (the minimum number of operations—insertions, deletions, substitutions—required to transform one string into the other.) If $\delta > 2$ is the threshold edit distance, take m_2 to be any message of length at least $\delta + 1$. Let m_0 be m_2 but with the first letter changed. Let m_1 be m_2 but with the last δ letters changed. Then m_0 and m_2 share more neighbors than m_1 and m_2 share, so these messages satisfy (1) and FSEtagNbs is IND-CLS-CPA-insecure in this case.

CONSTRUCTION OF THE FIRST SECURE EFSE SCHEME. We now improve on the scheme of [18] and construct an EFSE scheme that is IND-CLS-CPA-secure even on

[8] There are minor differences—notably, FSEtagNbs uses an IND-DCPA-secure ESE rather than a (stronger) IND-CPA-secure scheme, but this is not an issue as [18] leaks equality already through its encoding strategy. Moreover, we could instantiate FSEtagNbs with an IND-CPA-secure scheme in place of ESE and the attack described would still work, since the attack exploits the $\mathcal{F}_{\texttt{Ben}}$-tagged neighbors, not ESE. Other differences in [18] are inconsequential to the analysis.

rigid closeness domains. Let $\Lambda = (\mathcal{D}, \mathrm{Cl})$ be a closeness domain with $\mathcal{D}$ finite. Let $\mathcal{G}_\Lambda = (\mathcal{V}_\Lambda, \mathcal{E}_\Lambda)$ be the closeness graph of Λ. For $m \in \mathcal{D}$, let $E_m = \{\{m, m'\} \in \mathcal{E}_\Lambda \mid m' \in \mathcal{V}_\Lambda\}$ be the set of incident edges to m in $\mathcal{G}_\Lambda$, and note that message degree $\Delta_m = |E_m|$ and max degree $\Delta = \max_{m \in \mathcal{D}} \Delta_m$.

So that all messages have the same number of close neighbors, we introduce dummy messages. Construct a new graph $\mathcal{G}_{\mathtt{dum}} = (\mathcal{V}_{\mathtt{dum}}, \mathcal{E}_{\mathtt{dum}})$ where $\mathcal{V}_{\mathtt{dum}} = \mathcal{V}_\Lambda \cup \{w_1, \ldots, w_\Delta\}$, and $\mathcal{E}_{\mathtt{dum}}$ consists of all edges in $\mathcal{E}_\Lambda$, plus for any $m \in \mathcal{V}_\Lambda$, if $\Delta - \Delta_m > 0$ then let $\mathcal{E}_{\mathtt{dum}}$ also contain edges $\{m, w_1\}, \ldots, \{m, w_{\Delta - \Delta_m}\}$. We call these additional edges *dummy edges* and $w_1, \ldots, w_\Delta$ *dummy vertices*. $\mathcal{G}_{\mathtt{dum}}$ is thus a graph in which every element of $\mathcal{V}_\Lambda \subset \mathcal{V}_{\mathtt{dum}}$ has degree Δ.

Define $\mathtt{TagEdges} : \mathcal{D} \to \mathcal{E}_{\mathtt{dum}}$ as $\mathtt{TagEdges}(m) = \{e \in \mathcal{E}_{\mathtt{dum}} \mid m \in e\}$. Note: if $\mathrm{Cl}(m, m') = \mathtt{close}$ then $\mathtt{TagEdges}(m) \cap \mathtt{TagEdges}(m') \supseteq \{\{m, m'\}\} \neq \emptyset$; and if $\mathrm{Cl}(m, m') = \mathtt{far}$ then $\mathtt{TagEdges}(m) \cap \mathtt{TagEdges}(m') = \emptyset$. So $\mathtt{TagEdges}$ is a CPTF.

Let $\mathcal{F}_{\mathtt{Ben}}$ be a collision-free batch-encoding family on domain $\mathcal{E}_{\mathtt{dum}}$ and some range $\mathcal{R}_{\mathtt{En}}$, and let $\mathtt{ESE}$ be an ESE scheme on $\mathcal{D}$. Define the StructFSE scheme $\mathtt{FSEtagEdges}$ as $\mathtt{FSE}_{\mathtt{Etag}}[\mathtt{TagEdges}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ according to Figure 3. Notice that for all $m \in \mathcal{D}$, $|\mathtt{TagEdges}(m)| \leq \Delta$. So, if Λ has small max degree, $\mathtt{FSEtagEdges}$ is efficient.

Now, Theorem 4 provides the security guarantee of $\mathtt{FSEtagEdges}$. The proof is in [8], and simply shows the main condition of Theorem 2 (i.e., consistency of $\mathtt{TagEdges}$) is satisfied in this case.

Theorem 4. *If the max degree Δ of the closeness domain is small, and if* $\mathtt{ESE}$ *is IND-DCPA-secure and $\mathcal{F}_{\mathtt{Ben}}$ is PP-CBA-secure, then* $\mathtt{FSEtagEdges}$ *is IND-CLS-CPA-secure.*

Recall that certain blockcipher-based constructions (discussed earlier) satisfy the necessary efficiency, security, and functionality conditions for $\mathtt{ESE}$ and $\mathcal{F}_{\mathtt{Ben}}$. The final missing piece to achieve an efficient IND-CLS-CPA-secure scheme is that $\mathtt{TagEdges}$ should be efficient; i.e., for any message $m \in \mathcal{D}$ it should be easy to compute E_m. Thus, $\mathtt{FSEtagEdges}$ is an IND-CLS-CPA-secure EFSE scheme on Λ if the following two conditions hold:

(1) the max degree of Λ is small; (2) E_m is predetermined or calculated on-the-fly.

Of course, whether these conditions are satisfied depends on the closeness domain Λ. It is an interesting question to identify when (1) holds, and how to achieve (2) in those situations. However, the possibilities are wide-ranging and so we leave this as a topic of future research.

Now, we have successfully created a IND-CLS-CPA-secure scheme, but at what cost? It is apparent that, even if the max degree Δ is small enough for the scheme to be efficient, its size can lead to huge space-inefficiency, since ciphertexts in $\mathtt{FSEtagEdges}$ have length linear in Δ. And Δ could certainly be quite large—for instance, on a dense or high-dimensional metric closeness domain, even a small threshold supplies each message with many close neighbors.

Nevertheless, if we desire a general FSE construction to work on arbitrary closeness domains, such long ciphertexts are necessary. We explain in the following section.

LOWER BOUND ON CIPHERTEXT LENGTH OF AN FSE SCHEME FOR GENERAL CLOSENESS DOMAINS. Notice that our $\mathtt{FSEtagEdges}$ scheme is defined independently of the

closeness graph—in particular, the algorithms `makeDS` and `fuzzyQ` did not exploit any special structure of the closeness graph. In the following result, we show that to have such a scheme construction that is valid for "general" closeness domains, it requires ciphertext length linear in the max degree of the closeness domain. Moreover, note that this is an informational theoretic requirement, and relies only on functionality, rather than security, of the schemes. The proof of the theorem is in [8].

Theorem 5. *Let $\mathcal{D}$ be a fixed domain and Δ an integer with $2 \leq \Delta \ll |\mathcal{D}|$. There exists a family of closeness domains $\{\Lambda_i = (\mathcal{D}, \mathrm{Cl}_i)\}_{i \in I}$, each with max degree at most Δ, so that if $\{FSE_i\}_{i \in I}$ is a family of FSE schemes on the respective closeness domains that have common* `makeDS` *and* `fuzzyQ` *algorithms and a common ciphertext space, then the ciphertext length is at least $\Delta/2$.*

The bound on ciphertext length asymptotically matches the space-efficiency of scheme `FSEtagEdges` from the previous section, demonstrating that `FSEtagEdges` is "best-possible" for FSE schemes that work on general closeness domains.

6 Space-Efficient Schemes

Theorem 5 indicates that it is costly to construct EFSE schemes on general closeness domains. A natural question is whether we can improve efficiency by focusing on closeness domains that have nice structure. In particular, to avoid the strict conditions leading to Theorem 5 we should consider non-rigid closeness domains, where near message pairs enable "false positives" in a fuzzy query. However, note that if an adversary has any probabilistic edge in distinguishing near message pairs that lead to false positives and those that don't, he can easily break IND-CLS-CPA-security. To avoid such an attack, one must force the probability a near message pair is sent to a close ciphertext pair to be *uniform* over all near message pairs. But this negates the flexibility advantage of near messages—we expect an EFSE scheme satisfying this uniformity condition on near pairs would be as inefficient as the `FSEtagEdges` scheme. Thus, it appears that IND-CLS-CPA-security is too strong for more efficient EFSEs to achieve, even on non-rigid closeness domains. So to evaluate more efficient schemes, we need a new, weaker notion of security.

Intuitively, what information must a EFSE scheme on a non-rigid closeness domain Λ leak, given that some number of ciphertexts are known? Let H be the set of messages corresponding to known ciphertexts. For two messages in the same component of the induced nearness subgraph $\mathcal{G}_{\Lambda}^{\mathtt{N}}(H)$ (we say they are in the same *nearness component*) an EFSE is designed so that anyone might discover this fact by running `fuzzyQ` on their ciphertexts. So, by using EFSE we automatically give up a large amount of information about messages in the same nearness component (namely, their link through a chain of known near pairs.) It is a natural step to consider allowing more information leakage relating messages within the same nearness component, while protecting as much as possible about messages in different components, and hiding the "general location" of a message in the domain. We also might restrict our view to schemes on "regular" closeness domains—that is, domains where message closeness is defined in a similar manner in all parts of the space. Otherwise, irregularities in the domain would inherently reveal message locations.

Toward this end, we focus on real ℓ-dimensional domains where closeness of messages is defined regularly throughout the space. In particular, there is a regular lattice $\mathcal{L}$ such that the closeness function is invariant by $\mathcal{L}$-translations. Our new security notion then requires schemes to hide all information about plaintexts in different nearness components except for their "local structure" with respect to this lattice. The important implication is that nothing major (i.e., only "local structure") is revealed about the relationship between a pair of disconnected messages (i.e., messages that cannot be connected through a chain of near known corresponding ciphertext pairs). Hence, it is a sort of "macrostructure security" across disconnected nearness components.

In this section and related sections deferred to the full version [8], we focus on schemes achieving this security on certain metric closeness domains over $\mathbb{R}^\ell$. Suppose we can select a lattice $\mathcal{L} \subset \mathbb{R}^\ell$ and "anchor radius" $\rho > 0$ so that close messages are each within distance ρ of a common lattice point, and far messages are not. Then an obvious tagging strategy is to send a message to its *anchor points*: the lattice points within distance ρ of the message. We prove that the resulting scheme is secure with respect to $\mathcal{L}$ under the new definition. This new "macrostructure-secure" construction leads to a more detailed discussion that is relegated to the full version [8]. There, we pose an optimization problem related to the general construction, present some simple scheme constructions and a way to stitch simple constructions together to build useful schemes, then describe a practical instantiation of the scheme for fuzzy search on biometric data. Finally, in [8] we propose a direction of further research toward "probabilistic EFSE" schemes built out of locality-sensitive hash functions.

6.1 Macrostructure security on lattice-regular closeness domains

Our new notion of security will apply to closeness domains over $\mathbb{R}^\ell$ for which closeness is defined in a "regular" manner over the entire space. We characterize this regularity using a regular lattice on $\mathbb{R}^\ell$. Then, the security notion will hide everything about plaintexts except for how they locally relate to this regular lattice.

LATTICE-REGULAR CLOSENESS DOMAINS. Let $\mathcal{L}$ be a regular lattice in $\mathbb{R}^\ell$, that is, a set of vectors characterized as all integer combinations of a finite set of linearly independent basis vectors. We say a closeness domain $\Lambda = (\mathbb{R}^\ell, \mathrm{Cl})$ is *$\mathcal{L}$-regular* if for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^\ell$ and any $\mathbf{w} \in \mathcal{L}$, $\mathrm{Cl}(\mathbf{x}, \mathbf{y}) = \mathrm{Cl}(\mathbf{x} + \mathbf{w}, \mathbf{y} + \mathbf{w})$. That is, closeness relations are invariant under translation by any lattice vector. We say $\mathcal{L}$ is a *regularity lattice* of Λ. Also, if $\mathbf{z} = \mathbf{x} + \mathbf{w}$ for some $\mathbf{x}, \mathbf{z} \in \mathbb{R}^\ell$ and $\mathbf{w} \in \mathcal{L}$, we say that $\mathbf{x}$ and $\mathbf{z}$ are in the same *$\mathcal{L}$-class* and that $\mathbf{w}$ is the *$\mathcal{L}$-witness* from $\mathbf{x}$ to $\mathbf{z}$.

MACROSTRUCTURE SECURITY. Let $\mathcal{L}$ be a regular lattice on $\mathbb{R}^\ell$ and let $\Lambda = \left(\mathbb{R}^\ell, \mathrm{Cl}\right)$ be a $\mathcal{L}$-regular closeness domain on $\mathbb{R}^\ell$. The security notion is as follows.

Definition 3. *Let* $\mathtt{FSE} = (\mathcal{K}, \mathcal{E}nc, \mathcal{D}ec, \mathtt{makeDS}, \mathtt{fuzzyQ})$ *be an EFSE scheme on $\mathcal{L}$-regular closeness domain Λ. For an adversary A and $b \in \{0,1\}$, let* $\mathbf{Exp}^{\text{ind-nr}\mathcal{L}\text{-cpa-}b}_{\mathtt{FSE}}(A)$ *be identical to IND-CPA experiment* $\mathbf{Exp}^{\text{ind-cpa-}b}_{\mathtt{FSE}}(A)$ *in Figure 1, but with the restriction: for LR-queries (m_0^i, m_1^i), $i \in [q]$ made by the adversary, letting $H_0 = \{m_0^1, \ldots, m_0^q\}$ and $H_1 = \{m_1^1, \ldots, m_1^q\}$, require*

1. $|m_0^i| = |m_1^i|$ *for all* $i \in [q]$;

2. $\forall i \in [q]$, m_0^i *and* m_1^i *are in the same* $\mathcal{L}$*-class; furthermore, the* $\mathcal{L}$*-witness from* m_0^i *to* m_1^i *is also the* $\mathcal{L}$*-witness from* m_0^j *to* m_1^j *whenever* m_0^i *and* m_0^j *are in the same connected component of* $\mathcal{G}_{\Lambda}^{\mathtt{N}}(H_0)$.

For an adversary A*, define its* IND-NR$\mathcal{L}$-CPA advantage *against* FSE *as*

$$\mathbf{Adv}_{\mathtt{FSE}}^{\text{ind-nr}\mathcal{L}\text{-cpa}}(A) \;=\; \Pr\left[\mathbf{Exp}_{\mathtt{FSE}}^{\text{ind-nr}\mathcal{L}\text{-cpa-1}}(A) = 1\right] - \Pr\left[\mathbf{Exp}_{\mathtt{FSE}}^{\text{ind-nr}\mathcal{L}\text{-cpa-0}}(A) = 1\right].$$

We say that FSE *is* indistinguishable under same-nearness-component-$\mathcal{L}$-class chosen-plaintext attacks *(IND-NR$\mathcal{L}$-CPA-secure) or, alternatively,* macrostructure-secure with respect to anchor lattice $\mathcal{L}$ *(MacroStruct-$\mathcal{L}$-secure) if the IND-NR$\mathcal{L}$-CPA advantage of any adversary against* FSE *is small.*

The second LR-query requirement asks that a left-query component of $\mathcal{G}_{\Lambda}^{\mathtt{N}}(H_0)$ is a $\mathcal{L}$-translation (translation by a vector in $\mathcal{L}$) of the corresponding right-query component of $\mathcal{G}_{\Lambda}^{\mathtt{N}}(H_1)$. This implies that left and right queries have the same equality/closeness pattern, which we can see by the following. If $m_0^i = m_0^j$ then these messages are in the same nearness component (as they are the same vertex) so $\exists l \in \mathcal{L}$ with $m_1^i = m_0^i + l = m_0^j + l = m_1^j$. If $\text{Cl}(m_0^i, m_0^j) \in \{\mathtt{close}, \mathtt{near}\}$ then these messages are in the same nearness component so $\exists l \in \mathcal{L}$ with $m_1^i = m_0^i + l$, $m_1^j = m_0^j + l$, implying $d(m_1^i, m_1^j) = d(m_1^i + l, m_1^j + l) = d(m_0^i, m_0^j)$, so $\text{Cl}(m_1^i, m_1^j) = \text{Cl}(m_0^i, m_0^j)$. Thus, MacroStruct-$\mathcal{L}$-security is clearly weaker than IND-CLS-CPA-security.

Returning to the big picture, an MacroStruct-$\mathcal{L}$-secure scheme may leak how all messages in a nearness component lie with respect to nearby points in the regularity lattice. However, since the lattice itself is regular, no information is leaked about where those nearby lattice points actually are. Thus, for messages in different nearness components, an adversary learns nothing about the distance between them, or their approximate locations in the space, besides some bits with low significance, and that the distance is above $\delta^{\mathtt{F}}$ (which is by design.)

Practitioners should be aware that, depending on the application, MacroStruct-$\mathcal{L}$-security is not always an appropriate security guarantee. For instance, consider a scenario where IP addresses are encrypted by a MacroStruct-$\mathcal{L}$-secure scheme and the lattice points are IP addresses with the final byte equal to 0. The scheme could possibly leak the last byte of each IP address, perhaps revealing the particular types of conversants in IP traffic data. In general, when the "least significant" bits of data contain sensitive information, MacroStruct-$\mathcal{L}$-security may not be enough.

6.2 General macrostructure-secure construction on metric closeness domains

We aim to construct space-efficient EFSE schemes that meet our new notion of MacroStruct-$\mathcal{L}$-security for some regularity lattice. For practicality, we focus on the metric closeness domain on $\mathbb{R}^\ell$, Euclidean metric d, close threshold $\delta^{\mathtt{C}} > 0$, and far threshold $\delta^{\mathtt{F}} \geq \delta^{\mathtt{C}}$, i.e., $\Lambda = \left(\mathbb{R}^\ell, \mathcal{M}_d^{\delta^{\mathtt{C}}, \delta^{\mathtt{F}}}\right)$. Notice that Λ is $\mathcal{L}$-regular for *any* lattice $\mathcal{L} \subset \mathbb{R}^\ell$. We now define a few useful objects that will play a leading role in the general construction. Then, the construction follows.

ANCHOR RADII AND POINTS. Fix a lattice $\mathcal{L}$ in $\mathbb{R}^\ell$. For $\rho > 0$, we say that ρ is an *anchor radius* on closeness domain Λ and lattice $\mathcal{L}$, and $\{\mathbf{v} \in \mathcal{L} \mid d(\mathbf{m}, \mathbf{v}) \leq \rho\}$ is the set of *anchor points* of message $\mathbf{m}$, if (1) any two close messages $\mathbf{m}, \mathbf{m}' \in \mathcal{D}$ have a common anchor point, and (2) any two far messages $\mathbf{m}, \mathbf{m}' \in \mathcal{D}$ have no common anchor points.

GENERAL MACROSTRUCTURE-SECURE CONSTRUCTION AND ITS SECURITY. If ρ is an anchor radius on Λ and $\mathcal{L}$, then $\mathtt{TagsAnc}^\rho_\mathcal{L} : \mathbb{R}^\ell \to \mathcal{L}$ defined as $\mathtt{TagsAnc}^\rho_\mathcal{L}(\mathbf{m}) = \{\mathbf{v} \in \mathcal{L} \mid d(\mathbf{m}, \mathbf{v}) \leq \rho\}$ is a CPTF on Λ, as condition (1) implies that whenever $d(\mathbf{m}, \mathbf{m}') \leq \delta^{\mathtt{C}}$, there exists $\mathbf{v} \in \mathcal{L}$ such that $\mathtt{TagsAnc}^\rho_\mathcal{L}(\mathbf{m}) \cap \mathtt{TagsAnc}^\rho_\mathcal{L}(\mathbf{m}') \supseteq \{\mathbf{v}\}$; and condition (2) implies $\mathtt{TagsAnc}^\rho_\mathcal{L}(\mathbf{m}) \cap \mathtt{TagsAnc}^\rho_\mathcal{L}(\mathbf{m}') = \emptyset$ whenever $d(\mathbf{m}, \mathbf{m}') > \delta^{\mathtt{F}}$. Thus, if ρ is an anchor radius on Λ and $\mathcal{L}$, $\mathcal{F}_{\mathtt{Ben}} = (\mathcal{K}_{\mathtt{Ben}}, \mathtt{En}, \mathtt{Ben})$ is a collision-free batch-encoding family on domain $\mathcal{D}_{\mathtt{En}} = \mathcal{L}$, and $\mathtt{ESE}$ is an ESE scheme on $\mathcal{D}$, then the scheme $\mathtt{FSEtagAnc}^\rho_\mathcal{L} = \mathtt{FSE}_{\mathtt{Etag}}[\mathtt{TagsAnc}^\rho_\mathcal{L}, \mathcal{F}_{\mathtt{Ben}}, \mathtt{ESE}]$ is a StructFSE scheme by Theorem 1. The following result is proved in [8].

Theorem 6. $\mathtt{FSEtagAnc}^\rho_\mathcal{L}$ *defined ias above is MacroStruct-*$\mathcal{L}$*-secure provided* $\mathtt{ESE}$ *is IND-DCPA-secure,* $\mathcal{F}_{\mathtt{Ben}}$ *is PP-CBA-secure,* $\mu = \max_{\mathbf{m} \in \mathcal{D}} |\{\mathbf{v} \in \mathcal{L} \mid d(\mathbf{m}, \mathbf{v}) \leq \rho\}|$ *is small, and we can efficiently compute anchor points.*

Together, Theorem 1 and Theorem 6 say that if we can find an anchor radius ρ on closeness domain Λ and lattice $\mathcal{L}$ such that the maximum number of anchor points μ is small, and we can efficiently compute anchor points, $\mathtt{FSEtagAnc}^\rho_\mathcal{L}$ as constructed above is an MacroStruct-$\mathcal{L}$-secure EFSE scheme on Λ.

Note that the problem of finding a given message's anchor points is essentially the *ρ-close vectors problem (ρ-CVP)* on the appropriate parameters. Unfortunately, this problem is harder (assuming fixed maximum number of anchor points μ) then the standard closest vector problem with unlimited preprocessing, which has been shown to be NP-hard in general [19]. Thus, to ensure both efficiency and security in our specific constructions, it is vital to demonstrate how to efficiently compute anchor points.

The general "anchor-point" construction presented above provides a template for defining macrostructure-secure schemes. In the full version [8], we analyze some of the ramifications and possibilities. First, we pose the general open problem of how to choose anchor lattice and anchor radius to optimize space-efficiency and flexibility of a scheme. We next present several specific schemes, and identify how to stitch methods together to create a scheme supporting "conjunctive" closeness. Then, to enhance understanding, we describe and analyze a scheme for a practical application: supporting fuzzy search on biometric (fingerprint) data.

References

1. M. Adjedj, J. Bringer, H. Chabanne, and B. Kindarji. Biometric identification over encrypted data made feasible. In *Proceedings of the 5th International Conference on Information Systems Security*, ICISS '09, pages 86–100, Berlin, Heidelberg, 2009. Springer-Verlag.
2. G. Amanatidis, A. Boldyreva, and A. O'Neill. Provably-secure schemes for basic query support in outsourced databases. In S. Barker and G.-J. Ahn, editors, *DBSec*, volume 4602 of *Lecture Notes in Computer Science*, pages 14–30. Springer, 2007.

3. M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer, 2007.
4. M. Bellare, T. Kohno, and C. Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the Encode-then-Encrypt-and-MAC paradigm. *ACM Trans. Inf. Syst. Secur.*, 7(2):206–241, May 2004.
5. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.
6. M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In M. J. J. Jr., V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2009.
7. M. Bellare and P. Rogaway. On the construction of variable-input-length ciphers. In L. R. Knudsen, editor, *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 231–244. Springer, 1999.
8. A. Boldyreva and N. Chenette. Efficient fuzzy search on encrypted data. Full version of this paper. *Cryptology ePrint Archive* `https://eprint.iacr.org/`, 2013.
9. A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2009.
10. J. Bringer, H. Chabanne, and B. Kindarji. Error-tolerant searchable encryption. In *Proceedings of the 2009 IEEE international conference on Communications*, ICC'09, pages 768–773, Piscataway, NJ, USA, 2009. IEEE Press.
11. J. Bringer, H. Chabanne, and B. Kindarji. Identification with encrypted biometric data. *Security and Communication Networks*, 4(5):548–562, 2011.
12. Y.-C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In J. Ioannidis, A. Keromytis, and M. Yung, editors, *Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 442–455. Springer, 2005.
13. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 79–88. ACM, 2006.
14. C. Gentry. A fully homomorphic encryption scheme. *PhD Thesis, Stanford University*, 2009.
15. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.
16. M. Kuzu, M. S. Islam, and M. Kantarcioglu. Efficient similarity search over encrypted data. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 1156–1167, Washington, DC, USA, 2012. IEEE Computer Society.
17. A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In H. Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer, 2010.
18. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou. Fuzzy keyword search over encrypted data in cloud computing. In *INFOCOM*, pages 441–445. IEEE, 2010.
19. D. Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Transactions on Information Theory*, 47(3):1212–1215, 2001.
20. M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *J. Cryptology*, 12(1):29–66, 1999.

21. O. Pandey and Y. Rouselakis. Property preserving symmetric encryption. In *Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 375–391, Berlin, Heidelberg, 2012. Springer-Verlag.
22. P. Rogaway and T. Shrimpton. Deterministic authenticated-encryption: A provable-security treatment of the key-wrap problem. *IACR Cryptology ePrint Archive*, 2006.
23. D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.
24. V. Vaikuntanathan. Computing blindfolded: New developments in fully homomorphic encryption. In R. Ostrovsky, editor, *FOCS*, pages 5–16. IEEE, 2011.
25. C. Wang, Q. Wang, and K. Ren. Towards secure and effective utilization over encrypted cloud data. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops*, ICDCSW '11, pages 282–286, Washington, DC, USA, 2011. IEEE Computer Society.