Secure Conjunctive Keyword Search over Encrypted Data

Philippe Golle[1], Jessica Staddon[1], and Brent Waters[2]*

[1] Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
{pgolle,staddon}@parc.com
[2] Princeton University
Princeton, NJ 08544, USA
bwaters@cs.princeton.edu

Abstract. We study the setting in which a user stores encrypted documents (e.g. e-mails) on an untrusted server. In order to retrieve documents satisfying a certain search criterion, the user gives the server a *capability* that allows the server to identify exactly those documents. Work in this area has largely focused on search criteria consisting of a single keyword. If the user is actually interested in documents containing each of several keywords (*conjunctive* keyword search) the user must either give the server capabilities for each of the keywords individually and rely on an intersection calculation (by either the server or the user) to determine the correct set of documents, or alternatively, the user may store additional information on the server to facilitate such searches. Neither solution is desirable; the former enables the server to learn which documents match each individual keyword of the conjunctive search and the latter results in exponential storage if the user allows for searches on every set of keywords.

We define a security model for conjunctive keyword search over encrypted data and present the first schemes for conducting such searches securely. We propose first a scheme for which the communication cost is linear in the number of documents, but that cost can be incurred "offline" before the conjunctive query is asked. The security of this scheme relies on the Decisional Diffie-Hellman (DDH) assumption. We propose a second scheme whose communication cost is on the order of the number of keyword fields and whose security relies on a new hardness assumption.

Keywords: Searching on encrypted data.

1 Introduction

The proliferation of small hand-held devices and wireless networking enables mobile users to access their data at any time and from anywhere. For reasons

* Much of this work was completed while this author was an intern at PARC.

M. Jakobsson, M. Yung, J. Zhou (Eds.): ACNS 2004, LNCS 3089, pp. 31–45, 2004.

of cost and convenience, users often store their data not on their own machine, but on remote servers that may also offer better connectivity. When the server is untrusted, users ensure the confidentiality of their data by storing it encrypted.

Document encryption, however, makes it hard to retrieve data selectively from the server. Consider, for example, a server that stores a collection of encrypted emails belonging to a user. The server is unable to determine the subset of encrypted emails defined by a search criteria such as "urgent e-mail" or "e-mail from Bob".

The first practical solution to the problem of searching encrypted data by keyword is given in [15]. Documents and keywords are encrypted in a way that allows the server to determine which documents contain a certain keyword W after receiving from the user a piece of information called a *capability for keyword* W. The capability for W reveals only which documents contain keyword W and no other information. Without a capability, the server learns nothing about encrypted documents. Recent improvements and extensions to this scheme are given in [3,9,17].

A limitation common to all these schemes is that they only allow the server to identify the subset of documents that match a certain keyword, but do not allow for boolean combinations of such queries. Yet boolean combinations of queries appear essential to make effective use of a document repository, since simple keyword search often yields far too coarse results. For example, rather than retrieving *all* emails from "Bob", a user might only want those emails from Bob that are marked urgent and pertain to finance, in which case what is needed is the ability to search on the conjunction of the keywords, "Bob", "urgent" and "finance".

In this paper, we propose protocols that allow for *conjunctive* keyword queries on encrypted data. Although such conjunctive searches certainly do not encompass all possible search criteria, we believe that they are a crucial building block as indicated by the reliance of today's web search engines on conjunctive search (see, for example [10]). To motivate the problem of conjunctive search further, and illustrate the difficulties it raises, we briefly review two simple solutions and explain why they are unsatisfactory:

- **Set intersection.** A first approach to the problem of conjunctive keyword search is to build upon the simple keyword search techniques of [15]. Given a conjunction of keywords, we may provide the server with a search capability for every individual keyword in the conjunction. For every keyword, the server finds the set of documents that match that keyword, then returns the intersection of all those sets. This approach is flawed because it allows the server to learn a lot of extra information in addition to the results of the conjunctive query. Indeed, the server can observe which documents contain each individual keyword. Over time, the server may combine this information with knowledge of statistically likely searches to infer information about the user's documents.
- **Meta-keywords.** Another approach is to define a meta-keyword for every possible conjunction of keywords. Like regular keywords, these meta-keywords can be associated with documents. For example, a document that

contains the keywords "Bob", "urgent" and "finance" may be augmented with the meta-keyword "Bob: urgent: finance". With the techniques of [15], meta-keywords allow for conjunctive keyword search. The obvious drawback of this approach is that a document that contains m keywords requires an additional 2^m meta-keywords to allow for all possible conjunctive queries. This leads to an exponential (in m) blow-up in the amount of data that must be stored on the server.

These two failed approaches illustrate the twin requirements of conjunctive search protocols: security and efficiency. The first contribution of this paper is to formalize these goals. Specifically, we define a formal security model for conjunctive keyword search on encrypted data. This security model states, essentially, that the server should learn nothing other than the result of the conjunctive query. In particular, the server should not be able to generate new capabilities from existing capabilities, other than logical extensions, such as using a capability for W_1 and a capability for W_2 to generate a capability for $W_1 \wedge W_2$. Recall that security is only considered in the context of single keyword search in [3,15, 9], and so our definitions present a significant extension to prior security models.

We present two schemes that provably meet our definition of security. Both of our schemes come with a moderate storage cost. Our first scheme incurs a communication cost per query that is linear in the number of documents stored. However, the linear portion of this cost may be pre-transmitted and a constant size cost can then be paid when the user decides which query is of interest. Our second scheme works in groups for which there exists an admissible bilinear map [13,2] and relies on a new hardness assumption for its security. This scheme has the desirable attribute of requiring only constant communication with no need for pre-transmissions.

Overview. This paper is organized as follows. In Section 1.1 we discuss related work. Section 2 covers our notation, security definitions and hardness assumptions. We present a scheme for conjunctive search with amortized linear cost in Section 3 and a scheme with constant cost in Section 4. We conclude in Section 5.

1.1 Related Work

In [15], Song, Wagner and Perrig study a model of secure search over encrypted data that is similar to ours in that they consider a bandwidth constrained user who stores documents on an untrusted server. When the user needs all documents containing a certain keyword he provides the server with a small piece of information (called a *capability*) that enables the server to identify the desired (encrypted) documents. They propose an efficient, secret key method for enabling single keyword search that is provably secure. However, they do not provide a method for secure conjunctive search and it is hard to see how their techniques might be extended to accomplish this because their capabilities are deterministic and thus can potentially be combined to generate new capabilities. In our schemes we use modular exponentiation (hence, we incur more computational cost than [15]) and randomization of the capabilities to ensure that a

capability to search for documents containing both keyword W_1 and keyword W_2 is incompatible with a capability for W_1, and thus can't be used to generate a capability for W_2.

The use of search over encrypted data in file-sharing networks is investigated in [4], where a secret key system enabling sharing of, and searching for, encrypted data is described.

In [9], Goh presents an efficient scheme for keyword search over encrypted data using Bloom filters. Determining whether a document contains a keyword can be done securely in constant time, however, the scheme does not support secure conjunctive search.

The first public key schemes for keyword search over encrypted data are presented in [3]. The authors consider a setting in which the sender of an email encrypts keywords under the public key of the recipient in such a way that the recipient is able to give capabilities for any particular keyword to their mail gateway for routing purposes. Conjunctive keyword search is not supported in [3]. An efficient implementation of a public key scheme for keyword search tailored for documents that are the audit trails of users querying a database is in [17].

The related notion of negotiated privacy is introduced in [12]. A negotiated privacy scheme differs from the problem of encrypted search as studied here and in [15,3,9] in that the goal is to provide data collectors with the guaranteed ability to conduct specific searches.

Finally, we note that there are existing techniques for searching over encrypted data with increased security but with far less efficiency than our schemes and those described above. For example, private information retrieval (PIR) schemes (see, for example [6,7,5]) can potentially be used to solve this problem. A PIR scheme allows a user to retrieve information from a database server privately, that is without the server learning what information was retrieved. Hence, with a PIR scheme a user can search the documents stored on the database, and thus recover the documents of interest on their own. However, PIR schemes are designed in order to achieve higher security than we require (in a computational sense, the server in a PIR scheme has *no* information about what documents are retrieved) and thus come with far higher communication cost. Similarly, the notion of an oblivious RAM [11] can be leveraged to achieve heightened security, but with a significant efficiency cost. By accepting a weaker security guarantee that seems quite reasonable for our applications we are able to achieve a moderate communication cost.

2 Model

We consider a user that stores encrypted documents on an untrusted server. Let n be the total number of documents. We assume there are m keyword fields associated with each document. If documents were emails for example, we might define the following 4 keyword fields: "From", "To", "Date" and "Subject". For simplicity, we make the following assumptions:

– We assume that the same keyword never appears in two different keyword fields. The easiest way to satisfy this requirement is to prepend keywords

with the name of the field they belong to. Thus for example, the keyword "From:Bob" belongs to the "From" field and can not be confused with the keyword "To:Bob" that belongs to the "To" field.
- We assume that every keyword field is defined for every document. This requirement is easily satisfied. In our email example, we may assign the keyword "Subject:NULL" in the "Subject" field to emails that have no subject.

From here onwards, we identify documents with the vector of m keywords that characterize them. For $i = 1, \ldots, n$, we denote the ith document by $D_i = (W_{i,1}, \ldots, W_{i,m})$, where $W_{i,j}$ is the keyword of document D_i in the jth keyword field. The body of the ith document can be encrypted with a standard symmetric key cipher and stored on the server next to the vector of keywords D_i. For ease of presentation we ignore the body of the document and concern ourselves only with the encryption of the keyword vector, D_i.

When discussing a capability that enables the server to verify that a document contains a specific keyword in field j, we denote the keyword by W_j. A scheme for conjunctive keyword search consists of five algorithms, the first four of which are randomized:

- A parameter generation algorithm $\mathsf{Param}(1^k)$ that takes as input a security parameter k and outputs public system parameters ρ.
- A key generation algorithm $\mathsf{KeyGen}(\rho)$ that outputs a set K of secret keys for the user.
- An encryption algorithm $\mathsf{Enc}(\rho, K, D_i)$ that takes as input ρ, K and a document $D_i = (W_{i,1}, \ldots, W_{i,m})$ and outputs an encryption of the vector of keywords.
- An algorithm to generate capabilities $\mathsf{GenCap}(\rho, K, j_1, \ldots, j_\ell, W_{j_1}, \ldots, W_{j_\ell})$ that takes as input ρ, K as well as $1 \leq \ell \leq m$ keyword field indices $j_1, \ldots, j_\ell$ and ℓ keyword values $W_{j_1}, \ldots, W_{j_l}$ and outputs a value Cap, the *capability to search for keywords* $W_{j_1}, \ldots, W_{j_\ell}$. We call the portion of the capability that consists of the fields being searched over, $\{j_1, \ldots, j_\ell\}$, the support of the capability and denote it $\mathsf{Sup}(\mathsf{Cap})$.
- A verification algorithm: $\mathsf{Ver}(\rho, \mathsf{Cap}, \mathsf{Enc}(\rho, K, D_i))$ that takes as input ρ, a capability $\mathsf{Cap} = \mathsf{GenCap}(\rho, K, j_1, \ldots, j_\ell, W_{j_1}, \ldots, W_{j_\ell})$ and an encrypted document $\mathsf{Enc}(\rho, K, D_i)$ where $D_i = (W_{i,1}, \ldots, W_{i,m})$ and returns true if the expression $((W_{i,j_1} = W_{j_1}) \wedge (W_{i,j_2} = W_{j_2}) \wedge \ldots \wedge (W_{i,j_\ell} = W_{j_\ell}))$ holds and false otherwise.

Finally, throughout this paper we use the term *negligible function* to refer to a function $\eta : \mathbb{N} \to \mathbb{R}$ such that for any $c \in \mathbb{N}$, there exists $n_c \in \mathbb{N}$, such that $\eta(n) < 1/n^c$ for all $n \geq n_c$.

2.1 Security Definitions

A capability Cap enables the server to divide documents into two groups: those that satisfy the capability, and those that do not. Intuitively, a conjunctive keyword search scheme is secure if the server learns no other information from a

set of encrypted documents and capabilities. In this section, we formalize this notion of security. To facilitate the security definitions we define a randomized document $\text{Rand}(D, T)$, for any set of indices $T \subseteq \{1, \ldots, m\}$ and document $D = (W_1, \ldots, W_m)$. $\text{Rand}(D, T)$ is formed from D by replacing the keywords of D that are indexed by T (i.e., the set $\{W_i | i \in T\}$) by random values. Now we define *distinguishing capabilities*:

Definition 1. *A capability* Cap *is* distinguishing *for documents* D_i *and* D_j *if*

$$\mathsf{Ver}(\rho, \mathsf{Cap}, \mathsf{Enc}(\rho, K, D_i)) \neq \mathsf{Ver}(\rho, \mathsf{Cap}, \mathsf{Enc}(\rho, K, D_j))$$

Given a set of indices, $T \subseteq \{1, \ldots, m\}$*, a capability* Cap distinguishes *a document* D *from* $\text{Rand}(D, T)$ *if*

$$\mathsf{Ver}(\rho, \mathsf{Cap}, \mathsf{Enc}(\rho, K, D)) = \mathsf{true} \quad \textit{and} \quad T \cap \mathsf{Sup}(\mathsf{Cap}) \neq \emptyset$$

Note that with high probability the capabilities defined in part 2 of Definition 1 are distinguishing for D and $\text{Rand}(D, T)$ as defined in part 1 of the definition. We provide the second part of the definition largely to introduce some convenient terminology.

We define security for a conjunctive keyword search scheme in terms of a game between a polynomially bounded adversary $\mathcal{A}$ (the server) and a challenger (the user). The goal of $\mathcal{A}$ is to distinguish between the encryptions of two documents, D_0 and D_1 chosen by $\mathcal{A}$. Observe that $\mathcal{A}$ succeeds trivially if it is given a distinguishing capability for D_0 and D_1. We say that the scheme is secure if $\mathcal{A}$ cannot distinguish D_0 and D_1 with non-negligible advantage without the help of a distinguishing capability for D_0 and D_1. Formally:

Security Game ICC (indistinguishability of ciphertext from ciphertext)

1. The adversary, $\mathcal{A}$, adaptively requests the encryption, $\mathsf{Enc}(\rho, K, D)$, of documents, D, and search capabilities, Cap.
2. $\mathcal{A}$ picks two documents, D_0, D_1 such that none of the capabilities Cap given in step 1 is distinguishing for D_0 and D_1. The challenger then chooses b randomly from $\{0, 1\}$ and gives $\mathcal{A}$ an encryption of D_b.
3. $\mathcal{A}$ may again ask for encrypted documents and capabilities, with the restriction that $\mathcal{A}$ may not ask for a capability that is distinguishing for D_0 and D_1. The total number of all ciphertext and capability requests is polynomial in k.
4. $\mathcal{A}$ outputs $b_{\mathcal{A}} \in \{0, 1\}$ and is successful if $b_{\mathcal{A}} = b$. We define the adversary's advantage as: $Adv_{\mathcal{A}}(1^k) = |\Pr[b_{\mathcal{A}} = b] - 1/2|$, and the adversary is said to have an ϵ-advantage if $Adv_{\mathcal{A}}(1^k) > \epsilon$.

Definition 2. *We say a conjunctive search scheme is secure according to the game ICC if for any polynomial time adversary* $\mathcal{A}$*,* $Adv_{\mathcal{A}}(1^k)$ *is a negligible function of the security parameter* k*.*

We next define two variants of this security game that will simplify our proofs. In the first variant, the adversary chooses only one document D_0 as well as a subset T of the keywords of D_0. The challenger creates a document $D_1 = \text{Rand}(D_0, T)$. The goal of $\mathcal{A}$ is to distinguish between an encryption of D_0 and an encryption of D_1. As before, to make the game non-trivial, we need to place restrictions on the capabilities that $\mathcal{A}$ is allowed to ask for. Specifically, $\mathcal{A}$ may not ask for a capability that is distinguishing for D_0 and D_1.

Security Game ICR (indistinguishability of ciphertexts from random)

1. $\mathcal{A}$ may request the encryption $\mathsf{Enc}(\rho, K, D)$ of any documents D, and any search capabilities Cap.
2. $\mathcal{A}$ chooses a document D_0 and a subset $T \subseteq \{1, \ldots, m\}$ such that none of the capabilities Cap given in step 1 distinguishes D_0 from $D_1 = \text{Rand}(D_0, T)$. The challenger then chooses a random bit b and gives $\mathsf{Enc}(\rho, K, D_b)$ to $\mathcal{A}$.
3. $\mathcal{A}$ again asks for encrypted documents and capabilities, with the restriction that $\mathcal{A}$ may not ask for a capability that distinguishes D_0 from D_1. The total number of ciphertext and capability requests is polynomial in k.
4. $\mathcal{A}$ outputs $b_{\mathcal{A}} \in \{0, 1\}$ and is successful if $b_{\mathcal{A}} = b$. As in game ICC, we define the adversary's advantage as $Adv_{\mathcal{A}}(1^k) = |\Pr[b_{\mathcal{A}} = b] - 1/2|$.

Proposition 1. *If there is an adversary $\mathcal{A}$ that wins Game ICC with advantage ϵ, then there exists an adversary $\mathcal{A}'$ that wins Game ICR with advantage $\epsilon/2$.*

Proof. The proof of this proposition is standard and is left to the extended version of this paper.

Our final security game is quite similar to ICR except that we now consider an adversary who is able to distinguish between $\text{Rand}(D, T)$ and $\text{Rand}(D, T - \{t\})$, for some document D and set of indices T, $t \in T$. Again, this game enables simpler security proofs.

Security Game ICLR (indistinguishability of ciphertexts from limited random)

1. $\mathcal{A}$ may request the encryption $\mathsf{Enc}(\rho, K, D)$ of any documents D and any search capabilities Cap.
2. $\mathcal{A}$ chooses a document D, a subset $T \subseteq \{1, \ldots, m\}$ and a value $t \in T$ such that none of the capabilities Cap given in step 1 are distinguishing for $\text{Rand}(D, T)$ and $\text{Rand}(D, T - \{t\})$. The challenger then chooses a random bit b. If $b = 0$, the adversary is given $\mathsf{Enc}(\rho, K, D_0)$, where $D_0 = \text{Rand}(D, T - \{t\})$. If $b = 1$, the adversary is given $\mathsf{Enc}(\rho, K, D_1)$, where $D_1 = \text{Rand}(D, T)$.
3. $\mathcal{A}$ again asks for encrypted documents and capabilities, with the restriction that $\mathcal{A}$ may not ask for a capability that is distinguishing for D_0 and D_1. The total number of ciphertext and capability requests is polynomial in k.
4. $\mathcal{A}$ outputs $b_{\mathcal{A}} \in \{0, 1\}$ and is successful if $b_{\mathcal{A}} = b$. As in game ICC, we define the adversary's advantage as $Adv_{\mathcal{A}}(1^k) = |\Pr[b_{\mathcal{A}} = b] - 1/2|$.

Proposition 2. *If there is an adversary $\mathcal{A}$ that wins Game ICR with advantage ϵ, then there exists an adversary $\mathcal{A}'$ that wins Game ICLR with advantage ϵ/m^2.*

Proof. The proof of this proposition is standard and is left to the extended version of this paper.

2.2 Hardness Assumptions

The proofs of security of our conjunctive search schemes are based on two well-known hardness assumptions, Decisional Diffie-Hellman (DDH) and Bilinear Decisional Diffie-Hellman (BDDH). We briefly describe each of them here, referring the reader to [1] for additional information on DDH and to [2,13] for additional information on BDDH.

Decisional Diffie-Hellman. Let G be a group of prime order q and g a generator of G. The DDH problem is to distinguish between triplets of the form (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), where a, b, c are random elements of $\{1, \ldots, q-1\}$. We say a polynomial time adversary $\mathcal{A}$ has advantage ϵ in solving DDH if $|Pr[\mathcal{A}(g^a, g^b, g^{ab}) = \mathsf{true}] - Pr[\mathcal{A}(g^a, g^b, g^c) = \mathsf{true}]| > \epsilon$.

Bilinear Decisional Diffie-Hellman[1] Let G_1 and G_2 be groups of prime order q, with an admissible bilinear map (see [2]) $\hat{e} : G_1 \times G_1 \rightarrow G_2$, and let g be a generator of G_1. The BDDH problem is to distinguish 4-tuples of the form (g^a, g^b, g^c, g^{abc}) and (g^a, g^b, g^c, g^d), where a, b, c, d are random elements of $\{1, \ldots, q-1\}$. We say a polynomial time adversary $\mathcal{A}$ has advantage ϵ in solving BDDH if $|Pr[\mathcal{A}(g^a, g^b, g^c, g^{abc}) = \mathsf{true}] - Pr[\mathcal{A}(g^a, g^b, g^c, g^d) = \mathsf{true}]| > \epsilon$.

3 A Conjunctive Search Scheme with Constant Online Communication Cost

In the following protocol, the size of the capabilities for conjunctive queries is linear in the total number of documents stored on the server, but the majority of the communication cost between the user and the server can be done *offline*. More precisely, each capability consists of 2 parts:

- **A "proto-capability" part,** that consists of an amount of data that is linear in n, the total number of encrypted documents stored on the server. This data is *independent* of the conjunctive query that the capability allows, and may therefore be transmitted *offline*, possibly long before the user even knows the actual query that the proto-capability will be used for.
- **A "query" part:** a constant amount of data that depends on the conjunctive query that the capability allows. This data must be sent online at the time the query is made. Note that we call this amount of data *constant* because it does not depend on the number of documents stored on the server, but only on the number, m, of keyword fields per documents.

[1] BDDH has appeared in two forms, one in which the last element of the challenge 4-tuple is in the range of bilinear map and a stronger version that we present here and which is used in [16].

The following scenario illustrates how this search protocol might work in practice. An untrusted server with high storage capacity and reliable network connectivity stores encrypted documents on behalf of a user. Whenever the user has access to a machine with a high bandwidth connection (say a home PC), they precompute a lot of proto-capabilities and send them to the server. The server stores these proto-capabilities alongside the encrypted documents until they are used (proto-capabilities are discarded after being used once). If the user has only access to a low-bandwidth connection (a hand-held device for example) at the time they want to query their document repository, the user only need send the constant-size query part of the capability. The server combines that second part with one proto-capability received earlier to reconstitute a full capability that allows it to reply to the user's query. In this manner the high cost portion of the communication complexity can be pre-transmitted by the higher performance desktop and only a small burden is placed on the hand-held device.

Note that this scenario assumes the user does not store their documents directly on their own machine but on an untrusted server. We justify this assumption with the observation that the untrusted server likely offers more reliable and more available network connectivity than a machine belonging to the user.

System parameters and key generation. The function $\mathsf{Param}(1^k)$ returns parameters $\rho = (G, g, f(\cdot,\cdot), h(\cdot))$, where G is a group of order q in which DDH is hard, g is a generator of G, $f : \{0,1\}^k \times \{0,1\}^* \to \mathbb{Z}_q^*$ is a keyed function and h is a hash function. We use h as a random oracle. The security parameter k is used implicitly in the choice of the group G and the functions f and h. The key generation algorithm KeyGen returns a secret key $K \in \{0,1\}^k$ for the function f, and we denote $f(K,\cdot)$ by $f_K(\cdot)$. The family $\{f_K(\cdot)\}_K$ is a pseudorandom function family.

Encryption algorithm. We show how to compute $\mathsf{Enc}(\rho, K, D_i)$ where $D_i = (W_{i,1}, \ldots, W_{i,m})$. Let $V_{i,j} = f_K(W_{i,j})$ for $j = 1, \ldots, m$. Let a_i be a value chosen uniformly at random from $\mathbb{Z}_q^*$. The output is:

$$\mathsf{Enc}(\rho, K, D_i) = (g^{a_i}, g^{a_i V_{i,1}}, g^{a_i V_{i,2}}, \ldots, g^{a_i V_{i,m}})$$

Generating a capability $\mathsf{Cap} = \mathsf{GenCap}(\rho, K, j_1, \ldots, j_t, W_{j_1}, \ldots, W_{j_t})$**.** The capability Cap consists of a vector Q of size linear in the number of documents (the proto-capability that can be sent offline), and of an additional value of constant size (the query part). Let s be chosen uniformly at random from $\mathbb{Z}_q^*$. The vector Q is defined as:

$$Q = \left(h(g^{a_1 s}), h(g^{a_2 s}), \ldots, h(g^{a_n s})\right)$$

In addition, we define the value $C = s + (\Sigma_{w=1}^{t} f_K(W_{j_w}))$. The capability is the $(t+2)$-tuple, $\mathsf{Cap} = \{Q, C, j_1, \ldots, j_t\}$.

Verification. The server computes $R_i = g^{a_i C} \cdot g^{-a_i(\Sigma_{w=1}^{t}(V_{i,j_w}))}$ and returns true if $h(R_i) = h(g^{a_i s})$ and false otherwise.

3.1 Security Analysis

Proposition 3. *The scheme of Section 3 is secure according to game ICC in the random oracle model if DDH is hard in G.*

Proof. By Propositions 1 and 2, we know that the existence of an adversary that wins game ICC with non-negligible probability implies the existence of an adversary that wins game ICLR with non-negligible probability. Let $\mathcal{A}$ be an adversary that wins game ICLR with advantage ϵ. We build an adversary $\mathcal{A}'$ that uses $\mathcal{A}$ as a subroutine and breaks DDH with non-negligible advantage.

The algorithm $\mathcal{A}'$ first calls the function Param to generate the parameters $\rho = (G, g, f, h)$. Let g^a, g^b, g^c be a Diffie-Hellman challenge (the challenge is to determine whether $c = ab$). $\mathcal{A}'$ guesses a value z for the position t that $\mathcal{A}$ will choose in step 2 of the game ICLR, by picking z uniformly independently at random in $\{1, \ldots, m\}$.

The algorithm $\mathcal{A}'$ simulates the function Enc as follows. $\mathcal{A}'$ associates with every keyword W_i a random value x_i. When asked to compute $\mathsf{Enc}(\rho, k, D)$ where $D = (W_1, \ldots, W_m)$, $\mathcal{A}'$ chooses a random value a_i and outputs:

$$\mathsf{Enc}(\rho, k, D) = (g^{a_i}, g^{a_i x_1}, \ldots, (g^b)^{a_i x_z}, \ldots, g^{a_i x_m})$$

When asked to compute $\mathsf{Cap} = \mathsf{GenCap}(\rho, K, j_1, \ldots, j_t, W_{j_1}, \ldots, W_{j_t})$, $\mathcal{A}'$ outputs a vector $Q = (T_1, \ldots, T_n)$ of random values and a random value for C. To evaluate $\mathsf{Ver}(\rho, \mathsf{Cap}, \mathsf{Enc}(\rho, K, D_i))$, $\mathcal{A}$ must compute R_i and then ask $\mathcal{A}'$ for the value $h(R_i)$. $\mathcal{A}'$ knows whether D_i satisfies Cap or not. If it does, $\mathcal{A}'$ defines $h(R_i) = T_i$. Otherwise $\mathcal{A}'$ returns a random value for $h(R_i)$.

Finally, $\mathcal{A}$ submits a challenge document $D = (W_1, \ldots, W_m)$ for encryption along with a set $T \subseteq \{1, \ldots, m\}$ and a value $t \in T$. If $z \neq t$, $\mathcal{A}'$ returns a random guess in reply to the DDH challenge. With probability $1/m$, we have $z = t$ and in that case $\mathcal{A}'$ proceeds as follows. Let $E_t = (g^c)^{x_t}$. For $j \in T$, $j \neq t$, let $E_j = R_j$ for a random value R_j. For $j \notin T$, let $E_j = (g^a)^{x_j}$. $\mathcal{A}'$ returns to $\mathcal{A}$ the following ciphertext:

$$(g^a, E_1, \ldots, E_m)$$

Observe that this ciphertext is an encryption of D in every position $j \notin T$. If $c = ab$, this ciphertext is also an encryption of D in position t; otherwise it is not.

Now $\mathcal{A}$ is again allowed to ask for encryption of documents and for capabilities, with the restriction that $\mathcal{A}$ may not ask for capabilities that are distinguishing for $\mathrm{Rand}(D, T - \{t\})$ and $\mathrm{Rand}(D, T)$. This restriction ensures that $\mathcal{A}'$ can reply to all the queries of $\mathcal{A}$ as before.

Finally $\mathcal{A}$ outputs a bit $b_{\mathcal{A}}$. If $b_{\mathcal{A}} = 0$, $\mathcal{A}'$ guesses that g^a, g^b, g^c is not a DDH triplet. If $b_{\mathcal{A}} = 1$, $\mathcal{A}'$ guesses that g^a, g^b, g^c is a DDH triplet. Since the encryption will be random at position i if and only if the challenge is not a DDH tuple $\mathcal{A}'$ solves the DDH challenge with the same advantage that $\mathcal{A}$ has in winning game ICLR. □

4 A Conjunctive Search Scheme with Constant Communication Cost

In this section, we describe a protocol for which the total communication cost of sending a capability to the server is constant in the number of documents (but linear in the number of keyword fields). With this protocol, a low-bandwidth hand-held device will be able to construct capabilities on its own and the overall communication overhead will be low.

System parameters and key generation. The function $\mathsf{Param}(1^k)$ returns parameters $\rho = (G_1, G_2, \hat{e}, g, f(\cdot,\cdot))$, where G_1 and G_2 are two groups of order q, g is a generator of G_1, $\hat{e} : G_1 \times G_1 \to G_2$ is an admissible bilinear map and a keyed function $f : \{0,1\}^k \times \{0,1\}^* \to \mathbb{Z}_q^*$. The security parameter k is used implicitly in the choice of the groups G_1 and G_2. The key generation algorithm KeyGen returns a secret value α and K. Again, we denote $f(K,\cdot)$ by $f_K(\cdot)$, and $\{f_K(\cdot)\}_K$ forms a pseudorandom function family.

Encryption algorithm. We show how to compute $\mathsf{Enc}(\rho, K, D_i)$ where $D_i = (W_{i,1}, \ldots, W_{i,m})$. Let $V_{i,j} = f_K(W_{i,j})$ for $j = 1, \ldots, m$. Let $R_{i,j}$ for $j = 1, \ldots, m$ be m values drawn uniformly independently at random from $\mathbb{Z}_q^*$. Let a_i be a value chosen uniformly at random from $\mathbb{Z}_q^*$. The function Enc returns:

$$g^{a_i}, \left(g^{a_i(V_{i,1}+R_{i,1})}, \ldots, g^{a_i(V_{i,m}+R_{i,m})}\right), \left(g^{a_i \alpha R_{i,1}}, \ldots, g^{a_i \alpha R_{i,m}}\right)$$

Generating a capability $\mathsf{Cap} = \mathsf{GenCap}(\rho, K, j_1, \ldots, j_t, W_{j_1}, \ldots, W_{j_t})$**.** Let r be a value chosen uniformly at random from $\mathbb{Z}_q^*$. The capability Cap is:

$$\mathsf{Cap} = (g^{\alpha r}, g^{\alpha r(\sum_{w=1}^{t} f_K(W_{j_w}))}, g^r, j_1, \ldots, j_t)$$

Verification. We show how to compute $\mathsf{Ver}(\rho, \mathsf{Cap}, \mathsf{Enc}(\rho, K, D_i))$ where $\mathsf{Cap} = (g^{\alpha r}, g^{\alpha r(\sum_{w=1}^{t} f_K(W_{j_w}))}, g^r, j_1, \ldots, j_t)$ and $D_i = (W_{i,1}, \ldots, W_{i,m})$. The algorithm checks whether the following equality holds:

$$\hat{e}(g^{\alpha r(\sum_{w=1}^{t} f_K(W_{j_w}))}, g^{a_i}) = \prod_{k=1}^{t} \left(\frac{\hat{e}(g^{\alpha r}, g^{a_i(V_{i,j_k}+R_{i,j_k})})}{\hat{e}(g^r, g^{a_i \alpha R_{i,j_k}})} \right)$$

and returns true if the equality holds, and false otherwise.

4.1 Security Analysis without Capabilities

We first demonstrate a partial security result; namely, that when no capabilities are generated ciphertexts are indistinguishable provided BDDH is hard. To that end, we define a game ICC' which is identical to security game ICC of Section 2 except that *no* capabilities are generated (i.e. steps 1 and 3 are modified). Hence, the adversary who engages in Security Game ICC', renders an adaptive, chosen-plaintext attack.

Proposition 4. *If the Bilinear Decisional Diffie-Hellman (BDDH) problem is hard in G_1, then no adversary can win game ICC' with non-negligible advantage.*

Proof. Let $\mathcal{A}$ be an adversary who wins Security Game ICC' with advantage ϵ. We build an adversary $\mathcal{A}'$ which uses $\mathcal{A}$ as a subroutine and solves the BDDH problem. Let g^α, g^A, g^a, g^d be a BDDH challenge (the challenge is to decide whether $d = \alpha A a$).
When $\mathcal{A}$ asks for a document to be encrypted, $\mathcal{A}'$ does the following. For each keyword W_i it chooses a random value x_i. $\mathcal{A}'$ keeps track of the correspondence between keywords W_i and values x_i so that if a keyword appears multiple times (possibly in different documents), the same x_i is used consistently for that keyword. $\mathcal{A}'$ then chooses a random value a_i and random values $R_{i,1}, \ldots, R_{i,m}$. Finally, $\mathcal{A}'$ outputs

$$g^{a_i}, \left(g^{a_i(Ax_1+R_{i,1})}, \ldots, g^{a_i(Ax_m+R_{i,m})} \right) \left(g^{a_i \alpha R_{i,1}}, \ldots, g^{a_i \alpha R_{i,m}} \right)$$

Note that $\mathcal{A}'$ can compute all of these values since it knows a_i, x_j and the $R_{i,j}$. Note also that the above is a valid encryption of the document requested by $\mathcal{A}$. Now for its challenge, $\mathcal{A}$ asks for one more document D to be encrypted. The problem is for $\mathcal{A}$ to determine whether the encryption it receives from $\mathcal{A}'$ is an encryption of D or of a random document. $\mathcal{A}'$ chooses random values $b_1, \ldots, b_m$ and outputs

$$g^a, \left(g^{b_1}, \ldots, g^{b_m} \right), \left(g^{\alpha b_1 - d x_1}, \ldots, g^{\alpha b_m - d x_m} \right)$$

Note that $\mathcal{A}'$ can compute the value above and that if $d = \alpha A a$, the encryption above is an encryption of D. Otherwise it is an encryption of a random document. $\mathcal{A}$ outputs a guess as to whether it's been given an encryption of D or an encryption of a random document, and $\mathcal{A}'$ outputs the same guess as to whether $d = \alpha A a$ or not. Hence, just as in Proposition 3, if $\mathcal{A}$'s advantage in Security Game ICC' is ϵ, then the advantage of $\mathcal{A}'$ in solving BDDH is ϵ. □

4.2 Security Analysis with Capabilities

We present here a complete security analysis of the protocol of Section 4, including capabilities. Unfortunately, in a security model that includes capabilities (Game ICC), we do not know how to reduce the security of the protocol to a standard security assumption. Indeed, the breadth of applications for bilinear maps often necessitates new, nonstandard, hardness assumptions (see, for example [8]). We rely on the following new assumption:

Hardness Assumption (Game HA):
We define the following game. Let $\mathcal{G}$ be a group of order q, and let $g \in \mathcal{G}$ be a generator of $\mathcal{G}$. We assume the existence of an admissible bilinear map $\hat{e} : \mathcal{G} \times \mathcal{G} \to \mathcal{G}_2$. The game proceeds as follows:

1. We choose two random values $a, \alpha \in \mathbb{Z}_q^*$ and give $\mathcal{A}$, the adversary g^a and g^α.
2. $\mathcal{A}$ can request as many times as it wants and in any order the following:
   - **A variable.** Whenever $\mathcal{A}$ requests a new variable, we pick a random value $x_i \in \mathbb{Z}_q^*$ and give the adversary g^{x_i}.
   - **A product.** $\mathcal{A}$ specifies a subset $S = \{i_1, \ldots, i_k\}$ of variables. We pick a random value $r \in \mathbb{Z}_q^*$ and return to the adversary g^r, $g^{\alpha r}$ and $g^{\alpha r(x_{i_1}+\ldots+x_{i_k})}$.
3. $\mathcal{A}$ chooses two subsets T and T' of indices such that $T \cap T' = \emptyset$.
4. We give $\mathcal{A}$ the value $g^{a\alpha x_i}$ for all $i \in T'$. Next, we flip a bit b. If $b = 0$, we give the adversary the value $g^{a\alpha x_i}$ for all $i \in T$. If $b = 1$, we give the adversary g^{r_i} for a randomly chosen value $r_i \in \mathbb{Z}_q^*$ for all $i \in T$.
5. $\mathcal{A}$ outputs a bit $b_{\mathcal{A}}$.

We say that $\mathcal{A}$ wins game HA if the following two conditions hold:

- The adversary's guess is correct, i.e. $b_{\mathcal{A}} = b$.
- Let $S_1, \ldots, S_n$ be the list of sets requested by $\mathcal{A}$ in step 2 of the game HA. For any $i = 1, \ldots, n$, if $S_i \subseteq (T \cup T')$ then $S_i \cap T = \emptyset$.

Proposition 5. *If game HA is hard for $\mathcal{G}_1$, then no adversary can win the game ICC with non-negligible advantage.*

Proof. By Proposition 1, we know that the existence of an adversary who wins game ICC with non-negligible advantage implies the existence of an adversary who wins game ICR with non-negligible advantage. Let $\mathcal{A}$ be an adversary who wins game ICR with non-negligible advantage. We show how to construct an algorithm $\mathcal{A}'$ that uses $\mathcal{A}$ as a subroutine and wins game HA with non-negligible probability. The algorithm $\mathcal{A}'$ begins by asking for two values g^a and g^α (step 1 of game HA).

Next, we show how $\mathcal{A}'$ simulates the encryption function Enc for $\mathcal{A}$. When $\mathcal{A}$ wants a document encrypted, $\mathcal{A}'$ asks for a variable g^{x_i} for every new keyword W_i. The algorithm $\mathcal{A}'$ keeps track of the correspondence between keywords and values in $\mathcal{G}$ such that it can reuse values consistently if a keywords appears several times. To compute $\mathsf{Enc}(\rho, K, D)$ where $D = (W_1, \ldots, W_m)$, the algorithm $\mathcal{A}'$ chooses a random value a_i and m random values $R_1, \ldots, R_m$ and gives to $\mathcal{A}$:

$$g^{a_i}, \left((g^{x_1})^{a_i} g^{R_1}, \ldots, (g^{x_m})^{a_i} g^{R_m}\right), \left((g^\alpha)^{a_i R_1}, \ldots, (g^\alpha)^{a_i R_m}\right)$$

We show now how $\mathcal{A}'$ simulates capabilities for $\mathcal{A}$. Suppose that $\mathcal{A}$ asks for the following capability: $\mathsf{Cap} = \mathsf{GenCap}(\rho, K, j_1, \ldots, j_t, W_{j_1}, \ldots, W_{j_t})$. The algorithm $\mathcal{A}'$ asks for the values g^r, $g^{\alpha r}$ and $g^{\alpha r(x_{j_1}+\ldots+x_{j_t})}$ and outputs:

$$\mathsf{Cap} = \left(g^r, g^{\alpha r}, g^{\alpha r(x_{j_1}+\ldots+x_{j_t})}\right)$$

It is easy to verify that $\mathsf{Cap} = \mathsf{GenCap}(\rho, K, j_1, \ldots, j_t, W_{j_1}, \ldots, W_{j_t})$.

At some point, $\mathcal{A}$ chooses a challenge document $D = (W_1, \ldots, W_m)$ and a subset $T \subseteq \{1, \ldots, m\}$ (step 2 of game ICR). Without loss of generality, we assume that every keyword W_i has already appeared, i.e. $\mathcal{A}'$ already has a corresponding value g^{x_i}. If not, $\mathcal{A}'$ simply asks for the missing values g^{x_i}. The adversary $\mathcal{A}'$ defines $T' = \{1, \ldots, m\} \setminus T$.

Now $\mathcal{A}'$ chooses m new random values $y_1, \ldots, y_m$ and computes $g^{\alpha y_1}, \ldots, g^{\alpha y_m}$. Next, $\mathcal{A}'$ submits the sets T and T' as in step 3 of game HA. In return, $\mathcal{A}'$ gets values $g^{\delta_1}, \ldots, g^{\delta_m}$, where $\delta_j = a\alpha x_j$ for every $j \in T'$ and for $j \in T$, either $\delta_j = a\alpha x_j$ or δ_j is random (recall that the goal of $\mathcal{A}'$ is to distinguish between these two cases). Finally, $\mathcal{A}'$ gives to $\mathcal{A}$ the following value as the encryption of the challenge document D chosen by $\mathcal{A}$:

$$g^a, \left(g^{y_1}, \ldots, g^{y_m}\right), \left((g^{\alpha y_1}/g^{\delta_1}), \ldots, (g^{\alpha y_m}/g^{\delta_m})\right)$$

It is easy to verify that this is a correct encryption of the challenge document D in every position $j \notin T$, and in every position $j \in T$, it is either an encryption of W_j or an encryption of random. In such positions, it is up to the adversary $\mathcal{A}$ to guess which.

In step 3 of game ICR, $\mathcal{A}$ is again allowed to ask for encryption of documents and capabilities. We simulate these exactly as above.

In step 4 of game ICR, $\mathcal{A}$ outputs a bit $b_{\mathcal{A}}$. The adversary $\mathcal{A}'$ then outputs the same bit $b_{\mathcal{A}'} = b_{\mathcal{A}}$. Clearly, if $\mathcal{A}$ wins game ICR with non-negligible advantage, then $\mathcal{A}'$ guesses the bit correctly in game HA with the same non-negligible advantage. What remains to be shown is that the second condition for winning the game holds. That holds since whenever $\mathsf{Ver}(\rho, \mathsf{Cap}, \mathsf{Enc}(\rho, K, D)) = \mathsf{true}$ we must have that the set T was not queried on and therefore for any S that $\mathcal{A}'$ requests to construct a capability $S \cap T = \emptyset$. □

5 Conclusion and Open Problems

We have presented two protocols for conjunctive search for which it is provably hard for the server to distinguish between the encrypted keywords of documents of its own choosing. Our protocols allow secure conjunctive search with small capabilities. Our work only partially solves the problem of secure Boolean search on encrypted data. In particular, a complete solution requires the ability to do *disjunctive* keyword search securely, both across and within keyword fields.

An important issue that isn't addressed by our security games is the information leaked by the capabilities. In both of our protocols, the server learns the keyword fields that the capability enables the server to search. This alone may be enough to allow the server to infer unintended information about the documents. It would be interesting to explore solutions for the secure search problem that also protect keyword fields.

References

1. D. Boneh. *The decision Diffie-Hellman problem.* In Proceedings of the Third Algorithmic Number Theory Symposium, Lecture Notes in Computer Science, Vol. 1423, Springer-Verlag, pp. 48–63, 1998.
2. D. Boneh and M. Franklin. *Identity based encryption from the Weil pairing.* In *SIAM J. of Computing*, Vol. 32, No. 3, pp. 586-615, 2003.
3. D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano. *Searchable public key encryption.* To appear in Adances in Cryptology – Eurocrypt '04. Cryptology ePrint Archive, Report 2003/195, September 2003. http://eprint.iacr.org/2003/195/
4. K. Bennett, C. Grothoff, T. Horozov and I. Patrascu. *Efficient sharing of encrypted data.* In proceedings of ACISP 2002.
5. C. Cachin, S. Micali and M. Stadler. *Computationally private information retrieval with polylogarithmic communication.* In Advances in Cryptology – Eurocrypt '99.
6. B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan. *Private information retrieval.* In proceedings of FOCS '95.
7. B. Chor,N. Gilboa and M. Naor. *Private Information Retrieval by Keywords.* Technical report, TR CS0917, Department of Computer Science, Technion, 1997
8. Y. Dodis. *Efficient construction of (distributed) random functions.* In proceedings of the Workshop on Public Key Cryptography (PKC), 2003.
9. E. Goh. *Secure Indexes.* In the Cryptology ePrint Archive, Report 2003/216, March 16, 2004. http://eprint.iacr.org/2003/216/
10. Google, Inc. The basics of Google search. http://www.google.com/help/basics.html
11. O. Goldreich and R. Ostrovsky. *Software protection and simulation on oblivious RAMs.* In J. ACM, pp.431-473, 1996.
12. S. Jarecki, P. Lincoln and V. Shmatikov. *Negotiated privacy.* In the International Symposium on Software Security, 2002.
13. A. Joux. *The Weil and Tate pairings as building blocks for public key cryptosystems.* In Proceedings Fifth Algorithmic Number Theory Symposium, 2002.
14. A. Joux and K. Nguyen. *Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups.* In IACR ePrint Archive: http://eprint.iacr.org/2001/003/
15. D. Song, D. Wagner and A. Perrig. *Practical Techniques for Searches on Encrypted Data.* In Proc. of the 2000 IEEE Security and Privacy Symposium, May 2000.
16. V. Tô, R. Safavi-Naini and F. Zhang. *New Traitor Tracing Schemes Using Bilinear Map.* In 2003 ACM Workshop on Digital Rights Management (DRM 2003), October 27, 2003, The Wyndham City Center Washington DC, USA.
17. B. Waters, D. Balfanz, G. Durfee and D. Smetters. *Building an Encrypted and Searchable Audit Log.* In proceedings of NDSS 2004.