Worry-Free Encryption: Functional Encryption with Public Keys

Amit Sahai[*]
Department of Computer Science
University of California - Los Angeles
sahai@cs.ucla.edu

Hakan A. Seyalioglu[†]
Department of Mathematics
University of California - Los Angeles
hakan@math.ucla.edu

ABSTRACT

In this work, we put forward the notion of *Worry-Free Encryption*. This allows Alice to encrypt confidential information under Bob's public key and send it to him, without having to worry about whether Bob has the authority to actually access this information. This is done by encrypting the message under a *hidden* access policy that only allows Bob to decrypt if his credentials satisfy the policy. Our notion can be seen as a functional encryption scheme but in a public-key setting. As such, we are able to insist that even if the credential authority is corrupted, it should not be able to compromise the security of any honest user.

We put forward the notion of Worry-Free Encryption and show how to achieve it for any polynomial-time computable policy, under only the assumption that IND-CPA public-key encryption schemes exist. Furthermore, we construct CCA-secure Worry-Free Encryption, efficiently in the random oracle model, and generally (but inefficiently) using simulation-sound non-interactive zero-knowledge proofs.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public Key Cryptosystems

General Terms

Security, Algorithms

Keywords

Functional Encryption, Public Key Cryptography

1. INTRODUCTION

Consider the following scenario: As an employee with access to privileged information, you're caught off-guard by a co-worker's request for sensitive data. While he claims to have sufficient clearance, you don't want to risk unauthorized access. One approach to solve this problem would be to check the requester's credentials in a database or with an authority charged with validating credentials. However, this raises several problems of its own. Most fundamentally, it may be that your co-workers level of clearance is sensitive information in itself, and therefore any system utilizing a database storing users' credentials would be unacceptable (e.g. while you may have 'Top-Secret' clearance, you may not have the authority to know whether someone else does).

Your needs would be met (and your worries relieved) if you had an encryption scheme that guaranteed that your co-worker could only recover the data if he has the proper credentials. Then, you could simply encrypt the data with respect to an access policy appropriate to the data, and be sure that you did not give unauthorized access.

Defining and constructing a scheme with such a guarantee is the focus of this work. Informally, we require the following security guarantees:

- The scheme should be secure against eavesdroppers, satisfying usual notions of indistinguishability.
- The policy of a ciphertext should remain hidden, even to a user that can decrypt the ciphertext, except for the information that the user's credentials satisfy the policy. This requirement can be crucial in many settings: For example, if a naval vessel sends out an encrypted message with a policy limiting access to "Specialists in analyzing enemy submarine activity," then one could reasonably conclude that an enemy submarine was spotted just by observing the access policy.

  In fact, we will consider an even more general form of encryption where a general *function* f is encrypted. The recipient only learns $f(x)$, where x is an encoding of the credentials of the recipient. Nothing else about the function is revealed.
- A user's public key should leak no information about his credentials.
- Even if the certification authority (that validates credentials) is corrupted, it should not be able to compromise the security of any encryptions prepared for honest users.

[*] Research supported in part from NSF grants 0830803, 0627781, 0716389, 0456717, and 0205594, an equipment grant from Intel, and an Okawa Foundation Research Grant.

[†] Research supported in part by a NSF Graduate Research Fellowship.

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.

We shall call a scheme which provides the above guarantees of security a *Worry-Free Encryption Scheme*, since a sender does not need to worry about whether a recipient is authorized to obtain a message before sending it. Defining and constructing such a scheme is the focus of this work.

1.1 Related Work

Worry-Free Encryption is most closely related to two notions previously considered in the literature: functional encryption [17] and conditional disclosure of secrets [5, 1].

Functional encryption (with Attribute-Based Encryption as a special case) seeks to deal with a very similar setting to ours, but with notable differences that make the notions incomparable. Most fundamentally, functional encryption requires a greater degree of trust in a central authority: there must be a Key Generation Authority, which if corrupted, can then decrypt messages sent to all users in the system. Reducing trust in these systems is an area of much interest and current research [6]. Such central trust is avoided in Worry-Free Encryption, where even if the Certification Authority is corrupted, it will not be able to compromise messages encrypted to any honest user. At the same time, by having the Key Generation Authority, functional encryption is able to avoid the need for public keys. This feature allows functional encryption to be applicable to a number of important settings, such as searching on encrypted data [10] and secure cloud storage [8, 3] where the identities of recipients may not be known during encryption. We stress that Worry-Free Encryption is not designed to address these settings; in our setting, encryption is done with respect to a particular user's public key.

When we examine some of the most active areas of research in functional encryption, other important differences emerge. For example, in *Attribute Based Encryption* [17, 3, 7, 13] a user can only decrypt if his credentials satisfy a policy associated with the ciphertext[1]. However, the privacy of the *policy* on the ciphertext is not protected. As illustrated by the submarine example above, this can be problematic in many contexts. Furthermore, known results generally restrict policies to a fairly restrictive class.

For certain forms of functional encryption, (also known as *Predicate Encryption* [4, 10, 19, 18, 11, 13]) the privacy of the policy associated with a given ciphertext is a central issue. However, the functions that can be encrypted by the best current work are extremely limited, with the state of the art being the ability to check if a dot product of two vectors is zero. In contrast, as detailed below, for Worry-Free Encryption we will be able to handle all polynomial-time computable functions.

Our notion is also closely related to the notion of protocols for *conditional disclosure of secrets* [5, 1], where (for the two-party setting), a receiver is able to obtain the sender's secret if a fixed condition C is satisfied by the receiver's input. Such protocols have been typically considered for specific (usually algebraic) conditions C. In contrast, in our setting, this condition itself is both chosen by the sender and must remain secret from the receiver.

1.2 Results

Our work has two main contributions:

- We introduce the notion of *Worry-Free Encryption*.
- We provide several constructions:
    - ○ Our basic construction of Worry-Free Encryption works for arbitrary polynomial-time functions and is secure under chosen-plaintext attacks. The scheme requires only the existence of IND-CPA public-key encryption. The main ingredient is Yao's garbled circuits [20].
    - ○ We build a Worry-Free Encryption scheme that is secure against adaptive chosen-ciphertext attacks in the random oracle model. Our transformation is quite efficient, and the new scheme requires only one additional public-key operation over our basic scheme. No additional assumptions are needed. To achieve this goal, we give a novel method to prove the well-formedness of a *collection* of ciphertexts.
    - ○ Finally, we show that if non-interactive zero knowledge proofs for NP exist, there exists IND-CCA2 secure Worry-Free Encryption without random oracles.

Additionally, we consider the problem of a dishonest certification authority that is colluding with a dishonest user Alice. We stress that the standard definition of Worry-Free Encryption already guarantees that the security of messages sent to *other* parties cannot be compromised. However, there is another concern. What if (honest) Bob is sending Alice the encryption of some function f, expecting her to only be able to recover $f(x)$ where x is Alice's credentials. With no other security requirements, Alice and the authority together may be able to recover f completely, and thereby obtain information the Bob never intended any individual recipient to gain. To deal with this, we define a strengthening of our notion that guarantees that even when Alice and the authority collude, when Bob encrypts a message to Alice, nothing beyond $f(x')$ for some particular input x' will be learned by Alice and the authority. We obtain this higher level of security, under the assumption that a variant of one-round 1-out-of-2 Oblivious Transfer exists. Our extension is a natural one which at least one classical OT protocol [15, 1] satisfies.

2. PRELIMINARIES

Throughout the paper, we will use arrowed variables to denote vectors (e.g. $\vec{C}$, $\vec{v}$), $x[i]$ to denote the i^{th} bit of a string x, $\bar{b}$ to denote $b \oplus 1$ for a bit b and $[1, k]$ to denote all integers between 1 and k inclusive. We also use the notation $x \circ y$ to denote the concatenation of the strings x and y and $|g|$ to denote the size of a circuit g. A function is called negligible if it grows slower than any inverse polynomial in an implied parameter (usually λ, the security parameter), and non-negligible if it is not negligible. A probability will be said to be overwhelming if it is within a negligible additive factor of 1. We use $x \overset{\$}{\leftarrow} E$ to denote that x is chosen uniformly from the set E. $\mathbb{M}^{m \times n}$ is the set of m by n matrices with possibly null entries.

We will also make use of an existentially unforgeable signature scheme $\mathcal{S} = (KeyGen, Sign, Ver)$ which while not necessary to satisfy our security definitions, is required unless another method to ensure users only encrypt using public keys that the certification authority (CA) publishes (such as a secure database to store public keys). Furthermore,

[1] We specifically are defining Ciphertext-Policy ABE, there is also a notion of Key-Policy ABE where policies are associated with attributes are assigned to ciphertexts.

while we only explicitly state it for the Setup algorithm, we assume all parties have access to the security parameter λ and are restricted to running in polynomial time in this parameter.

2.1 Randomized Encodings

Our main construction will use *decomposable randomized encodings* heavily. A decomposable randomized encoding of $g : \{0,1\}^n \rightarrow \{0,1\}^k$ represented as a circuit of size polynomial in the security parameter λ will split g into $2n$ components: $([g]_{i,b} : i \in [1,n], b \in \{0,1\})$ such that for a given x, $([g]_{i,x[i]} : i \in [1,n])$ will suffice to reconstruct $g(x)$ but will computationally leak no other information about g except the size of the circuit (this problem can be addressed by padding). Randomized encodings as used in this paper can be constructed for any function g which can be represented as a polynomial size circuit by using garbled circuits [20].

Definition 2.1. [2] *A decomposable randomized encoding consists of a pair of algorithms E, the encoder, and D, the decoder, such that for security parameter λ:*

○ **Decomposability:** *For f a circuit from $\{0,1\}^n \rightarrow \{0,1\}^k$:*

$$E(f, 1^\lambda) \rightarrow ([f]_{i,b} \in \{0,1\}^m : i \in [1,n], b \in \{0,1\})$$

for m a function of $n, |f|, k, \lambda$ with $m = poly(\lambda)$ if $|f| = poly(\lambda)$.

○ **Correctness:** *For any string $x \in \{0,1\}^n$:*

$$D(([f]_{i,x[i]})_{i\in[1,n]}) = f(x)$$

with overwhelming probability if $n = poly(\lambda)$.

○ **Privacy:** *There exists a probabilistic polynomial time simulator S s.t. for any family of strings $\{x_\lambda\}_{\lambda\in\mathbb{N}}$ and circuits $\{f_\lambda : \{0,1\}^{n_\lambda} \rightarrow \{0,1\}^{k_\lambda}\}_{\lambda\in\mathbb{N}}$ with $|x_\lambda| = n_\lambda$, $|f_\lambda| = s_\lambda$:*

$$S(1^\lambda, n_\lambda, s_\lambda, f_\lambda(x_\lambda)) \equiv_c ([f_\lambda]_{i,x_\lambda[i]} : i \in [1, n_\lambda]),$$

where $E(f, 1^\lambda) \rightarrow ([f]_{i,b} : i \in [1,n], b \in \{0,1\})$ and $X_n \equiv_c Y_n$ denotes that for any polynomial size (non-uniform) circuit family A_n, $|\Pr[A_n(X_n) = 1] - \Pr[A_n(Y_n) = 1]|$ is negligible in λ if $n_\lambda, k_\lambda, s_\lambda = poly(\lambda)$.

Such an encoding is possible for circuits with information theoretic privacy with $O(|g|2^d)$ expansion where d is the depth of the evaluating circuit [9, 12]. With one way functions, it is possible with expansion $O(\lambda|g|)$ [20, 2] to encode any polynomial time function against p.p.t. adversaries (by applying garbled circuits to universal circuits[2]).

Notice that randomized encodings, by virtue of **Privacy**, satisfy a notion of **Indistinguishability** (by the transitivity of computational indistinguishability). Informally, this implies that if f_0 and f_1 are circuits of the same (polynomially bounded) size with $f_0(x) = f_1(x)$, $([f_z]_{i,x[i]} : i \in [1, |x|])$ for $z \in \{0,1\}$ are indistinguishable to polynomial size distinguishing circuits.

3. WORRY-FREE ENCRYPTION

We now will define the concept of *Worry-Free Encryption* along with the security guarantees placed on it.

Definition 3.1. *A Worry-Free Encryption scheme is a public key encryption scheme with credential authorization. It consists of six algorithms:* Setup, Pre, Auth, CheckAuth, Enc, Dec *with the following functionalities:*

- Setup$(1^\lambda) \rightarrow (PP, MSK)$ *The setup to generate public parameters and master secret key.*
- Pre$(x, PP) \rightarrow (\Sigma, SK)$ *The preprocessing stage performed by the user with credentials $x \in \{0,1\}^n$.*
- Auth$(\Sigma, x, MSK) \rightarrow PK_x$ *The authorization stage performed by the CA which takes as input the user's preprocessing information Σ and credentials x (which are verified out of the model) and outputs the public key PK_x (note x is not a part of PK_x and will be hidden).*
- CheckAuth(PK_x, Σ, x, PP) *This step will output $\perp$ iff the returned public key PK_x is incompatible with the preprocessing Σ. If the Auth step was performed correctly, it will always accept.*
- Enc$(f, PK_x, PP) \rightarrow C$ *Encrypts under PK_x, the circuit $f : \{0,1\}^n \rightarrow \{0,1\}^k$.*
- Dec$(Enc(f, PK_x, PP), SK_x, PP) \rightarrow f(x)$.

For our construction we will assume n is fixed on setup. Much like usual notions of encryption reveal the length of the message, across our security definitions we will assume that for two encryptions to be indistinguishable they must be encryptions of circuits of the same size with the same number of output bits. However, by padding, notice that one can easily make smaller circuits encrypted in a way that is indistinguishable from larger circuits. For simplicity, we will also assume for our construction that 'k', the output size, is fixed on setup – but note that this assumption is only for notational simplicity, and is not needed for security.

3.1 Security Definitions

If at any point in the security games, a functionality returns $\perp$, the experiment will terminate and return $\perp$. The first security requirement will be made concerning the function f. A scheme $\mathcal{W}$ is **Message Secure** if for any p.p.t. A_1, A_2, A_3, the probability the experiment below outputs 1 is less than 1/2 plus a negligible factor:

```
MessageSecure$_{A_1,A_2,A_3}(1^\lambda)$ :
    $\mathcal{W}.Setup(1^\lambda) \rightarrow (PP, MSK)$,
    $A_1(1^\lambda, PP) \rightarrow (x, \sigma_1)$,
    $\mathcal{W}.Pre(x, PP) \rightarrow (\Sigma, SK)$,
    $\mathcal{W}.Auth(\Sigma, x, MSK) \rightarrow PK_x$,
    $A_2(PK_x, \sigma_1) \rightarrow (f_0, f_1, \sigma_2)$,
    Return 0 unless $|f_0| = |f_1|$,
    $z \xleftarrow{\$} \{0,1\}$ , $\mathcal{W}.Enc(f_z, PK_x, PP) \rightarrow C$,
    $A_3(C, \sigma_2) \rightarrow g$,
    Return 1 iff $g = z$.
```

The second requirement is that a receiver with credentials x can only learn $f(x)$ and $|f|$ from an encryption of f. $\mathcal{W}$ is **Function Hiding** if for any p.p.t. A_1, A_2, A_3, the probability the below experiment outputs 1 is less than 1/2 plus a negligible factor:

[2] Not that in most implementations it is unnecessary to consider full universal circuits. In the 'Submarines' example given previously it would have been sufficient to instead take C a circuit which takes policies and attributes as inputs.

```
$\textsc{FunctionHiding}_{A_1,A_2,A_3}(1^\lambda)$ :
    $\mathcal{W}.Setup(1^\lambda) \to (PP, MSK)$,
    $A_1(1^\lambda, PP) \to (\Sigma, x, \sigma_1)$,
    $\mathcal{W}.Auth(\Sigma, x, MSK) \to PK_x$,
    $A_2(PK_x, \sigma_1) \to (f_0, f_1, \sigma_2)$,
    Return 0 unless $|f_0| = |f_1| \wedge f_0(x) = f_1(x)$,
    $z \xleftarrow{\$} \{0,1\}, \mathcal{W}.Enc(f_z, PK_x, PP) \to C$,
    $A_3(C, \sigma_2) \to g$,
    Return 1 iff $g = z$.
```

We will say a *Worry-Free Encryption* scheme has hidden credentials if PK_x leaks no information about x. Since the CA must have access to the user's credentials, this guarantee can only hold if the CA is honest. $\mathcal{W}$ has **Hidden Credentials** if for any p.p.t. A_1, A_2 the probability the below experiment outputs 1 is less than 1/2 plus a negligible factor:

```
$\textsc{HiddenCredentials}_{A_1,A_2}(1^\lambda)$ :
    $\mathcal{W}.Setup(1^\lambda) \to (PP, MSK)$,
    $A_1^{\mathcal{O}}(1^\lambda, PP) \to (x_0, x_1, \sigma)$,
    Return 0 unless $|x_0| = |x_1|$,
    $z \xleftarrow{\$} \{0,1\}, \mathcal{W}.Pre(x_z, PP) \to (\Sigma, SK)$,
    $\mathcal{W}.Auth(\Sigma, x_z, MSK) \to PK_{x_z}$,
    $A_2^{\mathcal{O}}(PK_{x_z}, \sigma) \to g$,
    Return 1 iff $g = z$.
```

where A_1, A_2 have oracle access to:

```
$\mathcal{O}(y)$ :
    $\mathcal{W}.Pre(y, PP) \to (\Sigma_y, SK_y)$,
    Return $(\mathcal{W}.Auth(\Sigma_y, y, MSK) = PK_y)$.
```

A major departure from previous schemes is that by assuming the public key infrastructure, we will actually be able to guarantee full security against the certification authority. $\mathcal{W}$ is **Malicious Authority Secure** if for any p.p.t. A_1, A_2, A_3, A_4 the probability the scheme below outputs 1 is less than 1/2 plus a negligible factor:

```
$\textsc{MaliciousAuthority}_{A_1,A_2,A_3,A_4}(1^\lambda)$ :
    $A_1(1^\lambda) \to (x, PP, \sigma_1)$,
    $\mathcal{W}.Pre(x, PP) \to (\Sigma, SK)$,
    $A_2(\Sigma, \sigma_1) \to (PK, \sigma_2)$,
    If $\mathcal{W}.CheckAuth(PK, \Sigma, x, PP) = \perp$
        Return 0,
    $A_3(\sigma_2) \to (f_0, f_1, \sigma_3)$,
    Return 0 unless $|f_0| = |f_1|$,
    $z \xleftarrow{\$} \{0,1\}, \mathcal{W}.Enc(f_z, PK_x, PP) \to C$,
    $A_4(C, \sigma_3) \to g$,
    Return 1 iff $g = z$.
```

Note that our security models do make the assumption that honest users use the public key output by the central authority to encrypt. This will be where the (PP, MSK) pair will be useful since we will be able to have the central authority sign all issued public keys using the signing key MSK to be verified under the verification key PP. Note that this signature step could be omitted if some other way of assuring the validity of public keys was present (such as a secure database that the CA uses to store the public keys). For simplicity we will assume all functions output by the adversary are in the function space accepted by the encryption scheme in our analysis.

3.2 Intuition Behind the Construction

The basic intuition for the construction follows: For each $i \in [1, n]$ the user will generate a public, secret key pair corresponding to 0 or 1 according to the bit of his credentials at the index i (in other words, the user generates $PK_{i,x[i]}, SK_{i,x[i]}$ for each i) and sends the corresponding public keys to the CA who will then fill in the blanks in the table $(PK_{i,b} : i \in [1, n], b \in \{0, 1\})$ to mask the user's credentials in the public key. Then an encrypter will generate an encoding of the circuit to be sent and encrypt each component $[f]_{i,b}$ under $PK_{i,b}$. This will guarantee that the user can only decrypt at indices which match the value of his credentials at this index, giving him access to $([f]_{i,x[i]} : i \in [1, n])$ allowing the user to reconstruct $f(x)$.

However, the above construction has a weakness, assume that the central authority stored $(SK_{i,\overline{x[i]}} : i \in [1, n])$ corresponding to indices of the public key which the CA generated. Then, the authority would be able to decrypt a ciphertext sent to the user in all indices $(i, \overline{x[i]})$, allowing him to recover $([f]_{i,\overline{x[i]}} : i \in [1, n])$ and reconstruct $f(\overline{x})$.

To fix this, the user will generate an additional key pair, $(PK_{n+1,0}\ SK_{n+1,0})$ not related to the user's credentials. The CA will then fill out the table for all indices up to n. For example, if $n = 3$, the public key[3] would be:

$$PK = \begin{pmatrix} PK_{1,0} & PK_{2,0} & PK_{3,0} & PK_{4,0} \\ PK_{1,1} & PK_{2,1} & PK_{3,1} & \end{pmatrix}.$$

Now, the encrypter will modify his message function $f : \{0,1\}^n \to \{0,1\}^k$ slightly. Instead of encoding f, he will encode $f' : \{0,1\}^{n+1} \to \{0,1\}^k$ defined as:

$$f'(x \circ 0) = f(x)\ ,\ f'(x \circ 1) = 0^k$$

and decompose it as $([f']_{i,b} : i \in [1, n+1], b \in \{0, 1\})$.

For each index of the public key, the encrypter will now encrypt $[f']_{i,b}$ under $PK_{i,b}$ discarding $[f']_{n+1,1}$ entirely. This guarantees that only the user has access to a full $n+1$ tuple.

Since we will use the transformation above frequently, for a fixed circuit $f : \{0,1\}^n \to \{0,1\}^k$ we will denote $f' : \{0,1\}^{n+1} \to \{0,1\}^k$ the transformed circuit above and,

$$T_\lambda(f) \to ([f']_{i,b} : i \in [1, n+1], b \in \{0, 1\})$$

the whole transformation. Furthermore, we will assume f is only used as a black box in constructing f' which will guarantee f' is constructed in such a way that only adds a fixed size to the circuit. Therefore we may assume $|f_0| = |f_1| \Rightarrow |f_0'| = |f_1'|$.

Define two sets corresponding to all indices of a public key and the indices the user with credentials x generated:

$$I = \{(i, b) : i \in [1, n], b \in \{0, 1\}\} \cup \{(n+1, 0)\},$$

$$J_x = \{(i, x[i]) : i \in [1, n]\} \cup \{(n+1, 0)\}.$$

[3] Recall that n does not correlate to any level of security, only the underlying access structure. An $n = 1$ is enough to express an 'Authorized' - 'Not-Authorized' access structure.

3.3 The Construction

We will now give our first *Worry-Free Encryption* scheme using an IND-CPA secure public-key encryption scheme $\mathcal{E} = (KeyGen, Enc, Dec)$, an existentially-unforgeable signature scheme $\mathcal{S} = (KeyGen, Sign, Ver)$ and a decomposable randomized encoding (E, D). Let $L(\lambda)$ be a polynomial upper bound on the credential size and the circuit size to be encrypted. If a check fails the functionality will return $\perp$.

All functionalities for each of our schemes will expect $x \in \{0,1\}^n$ and $f : \{0,1\}^n \rightarrow \{0,1\}^k$ where n and k are fixed on setup and return $\perp$ if this is not the case for an input. For notational convenience we will label the indices of a matrix in $\mathbb{M}^{n \times 2}$ as $((i,b) : i \in [1,n], b \in \{0,1\})$.

```
∘ $Setup(1^\lambda)$:
    $\mathcal{S}.KeyGen(1^\lambda) \rightarrow (VK, SignK)$,
    Return $(PP, MSK) = (VK, SignK)$.

∘ $Pre(x, PP)$:
    For $(i,b) \in J_x, \mathcal{E}.KeyGen(1^\lambda) \rightarrow (PK_{i,b}, SK_{i,b})$,
    Set $\Sigma = (PK_{i,b} : (i,b) \in J_x) \in \mathbb{M}^{n+1 \times 2}$,
    Set $\vec{SK} = (SK_{i,b} : (i,b) \in J_x)$,
    Return $(\Sigma, \vec{SK})$.

∘ $Auth(\Sigma = (PK_{i,b} : (i,b) \in J_x), x, MSK)$:
    Check only $J_x$ indices in $\Sigma$ are not null,
    For $(i,b) \in I \setminus J_x$, $\mathcal{E}.KeyGen(1^\lambda) \rightarrow (PK_{i,b}, SK_{i,b})$,
    Set $\vec{PK} = (PK_{i,b} : (i,b) \in I)$,
    $\mathcal{S}.Sign(\vec{PK}, SignK) \rightarrow \sigma$,
    Return $(\vec{PK}, \sigma)$.

∘ $CheckAuth((\vec{PK}, \sigma), \Sigma, x, PP)$:
    Check $\vec{PK} \in \mathbb{M}^{n+1 \times 2}$ with $(n+1, 1)$ index null,
    Check $\mathcal{S}.Ver_{VK}(\sigma, \vec{PK}) = \textsc{True}$,
    For $(i,b) \in J_x$ check $\vec{PK}$'s $(i,b)$ entry is $PK_{i,b}$ from $\Sigma$.

∘ $Enc(f, (\vec{PK}, \sigma), PP)$:
    Check $\mathcal{S}.Ver_{VK}(\sigma, \vec{PK}) = \textsc{True}$,
    $T_\lambda(f) \rightarrow ([f']_{i,b} : i \in [1, n+1], b \in \{0,1\})$,
    For $(i,b) \in I$, $\mathcal{E}.Enc_{PK_{i,b}}([f']_{i,b}) \rightarrow C_{i,b}$,
    Return $(C_{i,b} : (i,b) \in I)$.

∘ $Dec((C_{i,b} : (i,b) \in I), \vec{SK}, PP)$:
    For $(i,b) \in J_x$, $\mathcal{E}.Dec_{SK_{i,x[i]}}(C_{i,b}) \rightarrow [f']_{i,b}$,
    Return $D([f']_{i,b} : (i,b) \in J_x) = f(x)$.
```

We now begin with the proofs of security. Notice that *Malicious Authority Secure* implies *Message Secure* since the security requirement is the same, with the only modification being that the CA may be malicious in the former. Therefore, we do not prove *Message Security* separately. It will be useful to define an ordering on the elements of I (first by column, then by row), we can then refer to the 'first j' ordered pairs without ambiguity.

Theorem 3.2. *If $\mathcal{E}$ is a IND-CPA public-key encryption scheme and (E, D) is a secure decomposable randomized encoding, $\mathcal{W}$ is a CPA secure Worry-Free Encryption Scheme.*

Proof of Function Hiding: We define a sequence of hybrids $\mathcal{H}_j$. In the FunctionHiding experiment, recall that the $(i,b) \in I$ index of $\mathcal{W}.Enc(f, (\vec{PK}, \sigma), PP)$ is:

$$C_{i,b} = \mathcal{E}.Enc_{PK_{i,b}}([f']_{i,b}).$$

For H_j, instead of generating the first j elements of $I \setminus J_x$ as above in the challenge ciphertext, generate them as:

$$C_{i,b} = \mathcal{E}.Enc_{PK_{i,b}}(0^m).$$

Since for all replaced (i,b) (which are not in J_x), the public keys were generated by the experiment, by a standard hybrid argument we can conclude that if (A_1, A_2, A_3) has a non-negligible advantage in the FunctionHiding game (H_0), it also has a non-negligible advantage in H_n.

However, notice that in the experiment H_n, the challenge ciphertext depends only on the values $\{[f'_z]_{i,b} : (i,b) \in J_x\}$ where f_z is the challenge function. By the indistinguishability requirement of the encoding, and that $f'_1(x \circ 0) = f'_0(x \circ 0)$ and $|f'_1| = |f'_0|$, the probability (A_1, A_2, A_3) outputs z is no greater than $1/2$ plus a negligible function. Therefore, (A_1, A_2, A_3) can not have a non-negligible advantage in the experiment H_n if the randomized encoding satisfies indistinguishability, contradicting the previous assertion. □

Proof of Hidden Credentials: Notice the scheme satisfies the definition of hidden credentials information theoretically since if both user and CA are honest (as assumed in the experiment), all elements of the public key are drawn from the same distribution, independently of x. □

Proof of Malicious Authority Security: As in the proof of *Function Hiding* by going through a series of hybrids (recall for the experiment to not abort the indices in $\vec{PK}$ corresponding to elements of J_x were generated by the experiment), if (A_1, A_2, A_3, A_4) has a non-negligible advantage in the original experiment, it has a non-negligible advantage in the modified experiment where for all $(i,b) \in J_x$ the (i,b) component of the challenge ciphertext is instead generated as (removing dependence on the challenge function):

$$C_{i,b} = \mathcal{E}.Enc_{PK_{i,b}}(0^m).$$

Therefore, after the output of f_0, f_1 by A_3, the challenge ciphertext is publicly computable from: $([f'_z]_{i,\overline{x[i]}} : i \in [1,n])$. But notice this is the first n components of the encoding:

$$([f'_z]_{i,\overline{x[i]}} : i \in [1,n]) \cup ([f'_z]_{n+1,1}).$$

where f_z is the challenge function. Since $f'_0(\overline{x} \circ 1) = f'_1(\overline{x} \circ 1) = 0^k$ the above distribution is computationally indistinguishable for $z = 0$ or $z = 1$, contradicting the assumption that A_4 outputs z with non-negligible probability. □

4. CHOSEN-CIPHERTEXT SECURITY

In this section, we address the natural problem of providing a CCA2 Secure *Worry-Free Encryption* scheme. The first step is to precisely define what it means for a *Worry-Free Encryption* scheme to be CCA2 secure.

Notice that under a chosen ciphertext attack, the obvious notion of *Hidden Credentials* becomes unattainable since the adversary may simply ask for the user's decryption of an encryption of the identity function ($f(x) = x$ for all x), which gives the adversary access to the user's credentials. For *Function Hiding*, the attacker (a malicious user) already has the secret key and a decryption oracle doesn't add any functionality. The most natural setting in which to consider chosen-ciphertext attacks is therefore *Malicious Authority Security* (which implies *Message Security*). Recall that in the *Malicious Authority* security game we assume a malicious CA, the guarantee of indistinguishability should hold as long as the authority assigns a public key that is consistent with the user's pre-processing information.

Formally, to prove CCA2 security, (A_1, A_2, A_3, A_4) are given access to the oracle $\mathcal{O}$ below in the MALICIOUSAUTHORITY experiment:

```
$\mathcal{O}(C')$ :
    If queried by $A_4$ and $C = C'$ return $\perp$,
    Else return $\mathcal{W}.Dec(C', SK, PP)$.
```

4.1 CCA2 Security with Random Oracles

We now provide an efficient construction of CCA2 secure Worry-Free Encryption in the random oracle model.

Intuitively, the scheme will work similarly to our previous construction, however, due to the discrete components of the ciphertexts, we must safeguard against the adversary reusing parts of the challenge ciphertext in decryption queries. It is with this motivation that we use an additional invocation of the public-key encryption scheme (the keys of which we will label PK^*, SK^*) which will allow us to ensure that an adversary that makes a valid decryption query does not reuse parts of the challenge ciphertext. This can be viewed as a consistency check through all components of the ciphertext.

The random oracle proof model incorporates a function $H : \{0,1\}^* \to \{0,1\}^l$ that is modeled as a random function. The function is treated as an oracle, so any adversary must query H directly in order to have any information about the output $H(x)$ for any $x \in \{0,1\}^*$.

In our construction we will assume a IND-CPA secure encryption scheme with what we call, *unpredictable ciphertexts*, meaning that for any public key and any message, the probability that the ciphertext takes a fixed value should be negligible. We point out that this is satisfied by most schemes in the literature already and can be trivially obtained by concatenating randomness to the ciphertext not used during decryption. Let $L(\lambda)$ be a polynomial bound on the size of the credentials and the circuit components (the output of T_λ, which is polynomial in the circuit size). We use $Enc(M; R)$ to indicate that we run the encryption using R as randomness. For this scheme, we will let $H : \{0,1\}^* \to \{0,1\}^l$ where l is an upper bound on the bits of randomness $\mathcal{E}.Enc$ requires for the encryption of $L(\lambda)+m'$ and $2m'$ bit messages where $m' = \Theta(\lambda)$ (one can take $m' = \lambda$ for simplicity but this can be optimized, we will assume the encryption scheme can take as input more randomness than needed).

In this section, we will also assume $[f]_{i,b}$, an encoding component will begin with the index (i, b) in plaintext and we use $(i, b) \in [f]_{i,b}$ to denote that $[f]_{i,b}$ has this label. While this requires the simulator in the privacy guarantee to have x as input, it does not affect our usage or indistinguishability. We now define $\mathcal{R}$:

```
∘ $Setup(1^\lambda)$:
    $\mathcal{S}.KeyGen(1^\lambda) \to (VK, SignK) = (PP, MSK)$,
    Return $(PP, MSK)$.

∘ $Pre(x, PP)$:
    For $(i,b) \in J_x, \mathcal{E}.KeyGen(1^\lambda) \to (PK_{i,b}, SK_{i,b})$,
    $\mathcal{E}.KeyGen(1^\lambda) \to (PK^*, SK^*)$,
    Set $\Sigma = ((PK_{i,b} : (i,b) \in J_x), PK^*)$,
    Set $\vec{SK} = ((SK_{i,b} : (i,b) \in J_x), SK^*)$,
    Return $(\Sigma, \vec{SK})$.

∘ $Auth(\Sigma = ((PK_{i,b} : (i,b) \in J_x), PK^*), x, PP)$:
    For $(i,b) \in I \setminus J_x$,
        $\mathcal{E}.KeyGen(1^\lambda) \to (PK_{i,b}, SK_{i,b})$,
    Set[4] $\vec{PK} = ((PK_{i,b} : (i,b) \in I), PK^*)$,
    $\mathcal{S}.Sign(\vec{PK}, SignK) \to \sigma$,
    Return $(\vec{PK}, \sigma)$.

∘ $CheckAuth((\vec{PK}, \sigma), \Sigma, x, PP)$:
    Check $\mathcal{S}.Ver_{VK}(\sigma, \vec{PK}) =$ TRUE,
    For $(i,b) \in J_x$ check $(i,b)$ entry of $\vec{PK}$ is $PK_{i,b} \in \Sigma$.

∘ $Enc(f, (\vec{PK}, \sigma), PP)$:
    Check $\mathcal{S}.Ver_{VK}(\sigma, \vec{PK}) =$ TRUE,
    $T_\lambda(f) \to ([f']_{i,b} : i \in [1, n+1], b \in \{0,1\})$,
    For all $(i,b) \in I$: $r_{i,b} \xleftarrow{\$} \{0,1\}^{m'}$,
    Generate $r, r^* \xleftarrow{\$} \{0,1\}^{m'}$ ,
    For all $(i,b) \in I$:
        $C_{i,b} = Enc_{PK_{i,b}}([f']_{i,b} \circ r_{i,b}; H([f']_{i,b} \circ r_{i,b} \circ r))$,(†)
    Set $C^* = Enc_{PK^*}(r \circ r^*; H(\{C_{i,b}\}_{(i,b) \in I} \circ r \circ r^*))$,
    Return $((C_{i,b} : (i,b) \in I), C^*)$.

∘ $Dec((C_{i,b} : (i,b) \in I), C^*, \vec{SK})$:
    For $(i,b) \in J_x$, $\mathcal{E}.Dec_{SK_{i,b}}(C_{i,b}) \to ([f']_{i,b} \circ r_{i,b})$,
    Run $\mathcal{E}.Dec_{SK^*}(C^*) \to (r \circ r^*)$,
    For $(i,b) \in J_x$, check:
        $C_{i,b} = Enc_{PK_{i,b}}([f']_{i,b} \circ r_{i,b}; H([f']_{i,b} \circ r_{i,b} \circ r))$,
    Check $C^* = Enc_{PK^*}(r \circ r^*; H(\{C_{i,b}\}_{(i,b) \in I} \circ r \circ r^*))$,
    For $(i,b) \in J_x$, check $(i,b) \in [f']_{i,b}$,
    Return $D([f']_{i,b} : (i,b) \in J_x) = f(x)$.
```

THEOREM 4.1. *If $\mathcal{E}$ is a IND-CPA public-key encryption scheme with unpredictable ciphertexts and (E, D) is a secure decomposable randomized encoding, then $\mathcal{R}$ is an IND-CCA2 secure Worry-Free Encryption Scheme in the RO model.*

PROOF OF FUNCTION HIDING. We begin by showing that the above scheme is a usual *Worry-Free Encryption* scheme before addressing CCA2 security. Let (A_1, A_2, A_3) be an adversary with non-negligible advantage in the FUNCTION-HIDING security game for $\mathcal{R}$ where H is modeled as a ran-

[4] Format $\vec{PK}$ as a matrix in $\mathbb{M}^{n+1 \times 2}$ with $(n+1, 1)$ index null followed by PK^*.

dom oracle. We define two intermediary security games, the modified line in the scheme is marked with (†).

∘ Let FH1 denote the event FUNCTIONHIDING outputs 1 while interacting with (A_1, A_2, A_3).

∘ For all $(i, b) \in I \setminus J_x$ let the (i, b) component of the challenge ciphertext be generated instead as:

$$Enc_{PK_{i,b}}([f']_{i,b} \circ r_{i,b}; R_{i,b})$$

with $R_{i,b} \xleftarrow{\$} \{0,1\}^l$. Call the event this modified experiment outputs 1 with (A_1, A_2, A_3), FH2.

∘ For all $(i, b) \in I \setminus J_x$ let the (i, b) component of the challenge ciphertext be generated instead as:

$$Enc_{PK_{i,b}}(0^m \circ r_{i,b}; R_{i,b})$$

with $R_{i,b} \xleftarrow{\$} \{0,1\}^l$. Call the event this modified experiment outputs 1 with (A_1, A_2, A_3), FH3.

LEMMA 4.2. *$\Pr(\text{FH3}) \leq 1/2 + \nu(\lambda)$ for some ν negligible.*

PROOF. This follows directly from the computational indistinguishability requirement of decomposable encodings since the challenge ciphertext only depends on the $(i, b) \in J_x$ components of the encoding of the challenge function and $f_0(x) = f_1(x)$, $|f_0| = |f_1|$ for both challenge functions submitted by the adversary in FUNCTIONHIDING. □

LEMMA 4.3. *$|\Pr(\text{FH3}) - \Pr(\text{FH2})|$ is negligible.*

PROOF. The above follows directly by the IND-CPA security of *Enc* recall that for all indices where the ciphertext component is modified, the public key was chosen by the experiment, not the adversary. □

LEMMA 4.4. *$|\Pr(\text{FH2}) - \Pr(\text{FH1})|$ is negligible.*

PROOF. Notice that unless (A_1, A_2, A_3) queries H on input $[f'_z]_{i,b} \circ r^z_{i,b} \circ r^z$ where f_z is the challenge function and $r^z_{i,b}, r^z$ are the corresponding randomness generated during the challenge ciphertext query for some $(i, b) \in I \setminus J_x$, the view of the adversary is identically distributed in both games. Therefore, we can upper bound this difference by the probability that (A_1, A_2, A_3) queries $H([f'_z]_{i,b} \circ r^z_{i,b} \circ r^z)$ in the modified game of FH2 (since up to the point of the query, the view is identical, the probability such a query happens is the same in both experiments).

Assume that (A_1, A_2, A_3) queries $H([f'_z]_{i,b} \circ r^z_{i,b} \circ r^z)$ for some $(i, b) \in I$ with non-negligible probability in the relevant experiment for FH2. Consider the experiment where instead of $r^z_{i,b}$ being generated after the challenge index z is chosen, both $r^0_{i,b}$ and $r^1_{i,b}$ are generated and either $[f'_0]_{i,b} \circ r^0_{i,b}$ or $[f'_1]_{i,b} \circ r^1_{i,b}$ is encrypted based on the challenge index. The above adversary could then mount a distinguishing attack on $\mathcal{E}$ because the adversary's view would be independent of $r^{\overline{z}}_{i,b}$ and therefore the adversary could only query $H([f'_{\overline{z}}]_{i,b} \circ r^{\overline{z}}_{i,b} \circ r^{\overline{z}})$ with negligible probability. Therefore, one could guess the challenge index from this hash query and by embedding an instance of the IND-CPA security game of $\mathcal{E}$, break the scheme's IND-CPA security. □

REMARK: We have just shown $\mathcal{R}$ is *Function Hiding*. Notice it also satisfies *Hidden Credentials* information theoretically. It remains only to show it is *Malicious Authority Secure* under adaptive chosen-ciphertext attacks. In the same way that *Function Hiding* was proven, we can also show that the scheme is *Malicious Authority* secure against chosen-plaintext attacks, it remains only to argue that the scheme is *Malicious Authority* secure against CCA2 attacks.

PROOF OF CCA MALICIOUS AUTHORITY SECURITY. For simplicity from now on, we will always assume the users credentials x are set to 0^n. Since the user's credentials are not assumed private in this game, this does not effect the proof and allows the notation to be simplified. We begin by assuming $\mathcal{E}$ is an IND-CPA secure public key encryption scheme with *unpredictable ciphertexts.* As outlined above, malicious authority security under CPA queries can be shown similarly to the proof of function hiding. We now show that all ciphertext decryption queries in the CCA2 game can be simulated without ever having access to a decryption oracle (if the simulator controls H), which allows us to use the IND-CPA security of the scheme to achieve full CCA2 security.

By the property of *unpredictable ciphertexts*, the probability that an adversary makes a decryption query on a ciphertext which passes the decryption check without first querying H on the corresponding randomness of the ciphertext is negligible before seeing the challenge. Since the randomness in the ciphertext contains all information needed to decrypt, no decryption oracle is required to simulate decryption before the challenge ciphertext with overwhelming probability. However, this argument does not hold after the challenge ciphertext has been issued since the adversary may attempt to reuse part of the challenge ciphertext.

We now claim that (A_1, A_2, A_3) cannot, with non-negl. probability, make a ciphertext query that will pass the decryption check that re-uses either C^* or $C_{i,0}$ for some $i \in [1, n+1]$ of the challenge ciphertext (we call such a ciphertext *Overlapping.* We call a ciphertext *Valid* if it passes the check of the decryption step). For any decryption query $C = ((C_{i,b} : (i, b) \in I), C^*)$, there are two ways C could be overlapping (we will call the challenge ciphertext $\overline{C}$ and the corresponding ciphertext components $\overline{C^*}$, $\overline{C}_{i,0}$). For the following claims let $r^z_{i,b}, r^z, r^{*z}$ be the randomness used during encryption of the challenge query.

LEMMA 4.5. *Let C be any decryption query after the challenge phase with $C \neq \overline{C}$ and $C^* = \overline{C^*}$ such that no string containing r^z has been queried to H by the adversary. Then, the probability C is* valid *is negligible.*

PROOF. Since $C \neq \overline{C}$, $C_{i,b} \neq \overline{C}_{i,b}$ for some $(i, b) \in I$. Then, in order for C to be *valid*, it is necessary that:

$$\mathcal{E}.Enc_{PK^*}(r^z \circ r^{*z}; H(\{C_{i,b}\}_{(i,b)\in I} \circ r^z \circ r^{*z})) =$$
$$\mathcal{E}.Enc_{PK^*}(r^z \circ r^{*z}; H(\{\overline{C}_{i,b}\}_{(i,b)\in I} \circ r^z \circ r^{*z})).$$

However, note that the probability of the above equality is negligible by the unpredictable ciphertexts requirement of $\mathcal{E}$ because $H(\{C_{i,b}\}_{(i,b)\in I} \circ r^z \circ r^{*z})$ has not been queried. Therefore, with overwhelming probability C^* is not a valid encryption of the underlying message and C is not *valid.* □

LEMMA 4.6. *Let C be any decryption query after the challenge phase with $C \neq \overline{C}$, $C^* \neq \overline{C^*}$ such that $C_{j,0} = \overline{C}_{j,0}$ for some $j \in [1, n+1]$. If no string containing $r^z_{j,0}$ for any $j \in [1, n+1]$ or r^z has been queried to H by the adversary, the probability C is* valid *is negligible.*

PROOF: Since $C^* \neq \overline{C^*}$, if for some $s, s^* \in \{0,1\}^{m'}$,

$$C^* = \mathcal{E}.Enc_{PK^*}(s \circ s^*; H(\{C_{i,b}\}_{(i,b)\in I} \circ s \circ s^*))$$

then with overwhelming probability we may assume A has queried $H(\{C_{i,b}\}_{(i,b)\in I}, s, s^*)$ by the property of unpredictable ciphertexts. By our assumption this implies $s \neq r^z$. For this ciphertext to be valid, it is necessary that:

$$C_{j,0} = \mathcal{E}.Enc_{PK_{j,0}}([f'_z]_{j,0}, r^z_{j,0}; H([f'_z]_{j,0}, r^z_{j,0}, s))$$

However, since $r^z_{j,0}$ has not been queried as a component in H, the argument for H has not yet been queried and therefore the above will hold with only negligible probability by the guarantee of unpredictable ciphertexts for $\mathcal{E}$. □

Notice that if there is no *overlapping valid* ciphertext we can simply use the queries to the hash function to decrypt. Assume $C_{j,0} \neq \overline{C}_{j,0}$, then, if C is valid, there is some underlying $[f]_{j,0}, r_{j,0}, r$ (for the r implicit from the C^* location):

$$C_{j,0} = \mathcal{E}.Enc_{PK_{j,0}}([f']_{j,0}, r_{j,0}, H([f']_{j,0}, r_{j,0}, r))$$

such that $(j, 0) \in [f']_{j,0}$. If the adversary has not queried the hash argument, and the above was not the encryption of the $(j, 0)$ index of the challenge ciphertext (since this is the only index with $(j, 0) \in [f']_{j,0}$), the hash function has not been queried on this argument and the probability the above equality will hold is negligible. Note that decryption only relies on $C_{j,0}$ for $j \in [1, n + 1]$ and therefore reusing $C_{j,1}$ from the challenge ciphertext does not effect our ability to decrypt. Therefore a simulator can return $\perp$ on all decryption queries for overlapping ciphertexts and be accurate with overwhelming probability. We call the event in which the adversary queries a string with either $r^z_{i,0}$ for some $i \in [1, n + 1]$, r^z or r^{*z} as a substring to H an *oracle failure.* We showed above that to query an *overlapping valid* ciphertext, with overwhelming probability, the adversary must cause an *oracle failure.*

LEMMA 4.7. *If (A_1, A_2, A_3) causes an oracle failure with non-negligible probability, $\mathcal{E}$ is not IND-CPA secure.*

PROOF: We only address when the an oracle failure by querying a string containing $r^z_{i,0}$ (the other two cases follow similarly). Notice that as long as there have been no *oracle failures* the we can simulate the adversary's decryption queries with its queries to the hash function. Therefore, using (A_1, A_2, A_3) as a subroutine, we can make a second adversary that creates an *oracle failure* without ever using a decryption oracle. Let (A'_1, A'_2, A'_3) be such a tuple.

Up to the point of creating an *oracle failure*, the view of (A'_1, A'_2, A'_3) is identical if instead of using H to generate the randomness used during the encryption of the challenge ciphertext, this randomness is generated uniformly at random. Therefore, we may assume (A'_1, A'_2, A'_3) creates an *oracle failure* when the randomness used in the challenge ciphertext is drawn uniformly at random.

The proof now follows similarly to the case in *Function Hiding*, assume both $r^0_{i,b}$ and $r^1_{i,b}$ are created during the challenge encryption and $[f'_z]_{i,b}, r^z_{i,b}$ is encrypted with uniform randomness. If A' has non-negligible probability of querying the oracle on some $r^d_{i,b}$ this creates a distinguishing attack against the encryption scheme since its view is independent of $r^{\overline{z}}_{i,b}$ where z is the index chosen in the challenge query. □

4.2 CCA Security in the Standard Model

In this section we describe the construction of a CCA2 secure Worry-Free Encryption scheme in the standard model using our IND-CPA scheme. The construction follows heavily Sahai's construction of IND-CCA2 secure public key encryption from an IND-CPA secure scheme and a simulation sound NIZK proof system for NP [16].

In Sahai's original construction, the main observation is that if the IND-CPA secure scheme is run twice in parallel and a valid encryption includes a proof that both parallel encryptions are of the same message, one can transform an IND-CPA secure scheme to an IND-CCA2 secure public key encryption scheme. The application to our setting is nearly immediate, with the proof being that each pair of ciphertext components with the same index (i, b) are encryptions of the same message, with only the following caveat. For the NIZK proof system, it is necessary to have a common reference string of randomly generated bits. This is not a problem in Sahai's construction since the user can randomly generate the CRS and use it as part of its public key, a malicious user is not an issue. However, in our setting, in certain security games, this user may be dishonest and thus we can not trust that the CRS would be correctly generated if we allowed the user to generate it independently.

It is for this reason that we must use a coin flipping protocol (implemented through a commitment scheme) between the user and key generation authority in order to settle on this CRS, which allows the simulator to rewind during the proof to set the CRS. Since either the user or key generation authority is assumed honest in all security games for standard Worry-Free Encryption, the CRS will always be correctly generated and we will be able to set the CRS in each security experiment. However, this will be incompatible with the ideas in the following section in constructing a minimally-vulnerable scheme and we leave the problem of constructing a CCA2 secure Worry-Free encryption scheme that is minimally-vulnerable open.

5. MALICIOUS AUTHORITY COLLUSION

One question that arises is the case of a malicious user Alice and the CA colluding to make her public key. Since credentials in *Worry-Free Encryption* are hidden in the public keys, an honest Bob can never tell if Alice and the CA collude to make her a public key with incorrect credentials. However, in this section we will give a scheme where this is all they can accomplish. We call such a scheme *minimally vulnerable to collusion.*

5.1 Two Round 1-out-of-2 Oblivious Transfer

Oblivious transfer is a well developed concept in cryptographic literature [15, 14] and we will assume some proficiency in this paper. A two round (sometimes called 'non-interactive') 1-out-of-2 oblivious transfer protocol (OT^2_1) is a two round protocol between a chooser (who sends the first message) and a sender (who sends the second) such that the sender starts with two values M_0 and M_1 and if both parties are honest, at the end, the chooser will receive M_b for his choice of b and will gain no additional information about $M_{\overline{b}}$ apart from M_b and the sender gains no information about b.

However, for our purposes, we will need a slight modification on this concept. While not implied by the traditional definition, we conjecture it is satisfied by many known OT^2_1

constructions, one of the best known constructions, due to Naor & Pinkas [15] in particular satisfies this notion.

5.2 Static OT_1^2

We will now begin with defining the precise requirements we will require from the oblivious transfer protocol and label this new primitive *Static* OT_1^2. We formally define this primitive below.

A *Static* OT_1^2 protocol is between two parties, with one party, the *sender* having two inputs (M_0, M_1) and the other party the *chooser* has input one bit σ. The protocol is two rounds, with the first from *chooser* to *sender* and the second a response from *sender* to *chooser* such that at the end the *chooser* learns M_σ with the following guarantees:

Chooser Security: The *sender*'s view when $\sigma = 0$ or when $\sigma = 1$ are computationally indistinguishable.

Sender Security: We use the *ideal implementation* definition where a trusted third party receives M_0 and M_1 from the sender and σ from the chooser and returns M_σ to the chooser. For any distribution on (M_0, M_1) and any polynomial time adversarial chooser $\mathcal{A}$ in the real implementation, there exists a simulator $\mathcal{A}'$ that takes the chooser's role in the ideal model with the same inputs as $\mathcal{A}$ such that the outputs of $\mathcal{A}$ and $\mathcal{A}'$ are indistinguishable given M_0, M_1.

Static Retrieval: For any first round message from an adversarial chooser there is a bit β such that any response by a honest sender reveals no information about M_β.

5.3 Security Definition

Intuitively, the definition we desire in this situation is clear. Assume the user and CA collude to make a public key PK. By the requirement of *hidden credentials* it should be impossible for an honest user to determine the set of credentials associated with PK, if there even is one. Perhaps this collusion makes it possible to recover more about the function than evaluation at a point.

The security requirement we would like is that every maliciously generated PK has with it associated some x such that all that can be recovered from $Enc(f, PK, PP)$ is $f(x)$. However, there is a slight difficulty in defining this since the malicious users may not be aware of what this implied credential x is, if such a credential even exists. Therefore, we will make use of *Ext* an exponential time extractor which recovers such an x. Recall that the guarantees on decomposable randomized encodings hold against non-uniform circuit families; this will be crucial in maintaining security even if the challenge function is generated after seeing the output of this exponential time extractor.

We define the relevant experiment for our purposes below.

```
$\mathrm{EXPT}_{A_1,A_2,A_3}(1^\lambda)$:
    $A_1(1^\lambda) \rightarrow (PK, PP, \sigma_1)$,
    $Ext(PK) \rightarrow x \in \{0,1\}^n$,
    $A_2(PK, x, \sigma_1) \rightarrow (f_1, f_2, \sigma_2)$,
    $z \xleftarrow{\$} \{0,1\}$, $\mathcal{W}.Enc(f_z, PK, PP) \rightarrow C$,
    $A_3(C, \sigma_2) \rightarrow g$,
    Return $(g, f_0, f_1, z)$.
```

We will say for the preceding experiment that a scheme is **Minimally Vulnerable to Collusion** if there exists a possibly exponential time deterministic extractor *Ext* such that:

$$\Pr[g = z \wedge f_0(x) = f_1(x) \wedge |f_0| = |f_1|] \leq 1/2 + \nu(\lambda)$$

where ν is negligible. Our scheme is below where $\mathcal{O}$ is a Static OT_1^2 protocol with rounds $\mathcal{O}_1$, $\mathcal{O}_2$ with d the state from $\mathcal{O}_1$ that allows reconstruction by the function R. In other words, the steps of the scheme can be represented as $\mathcal{O}_1(b) \rightarrow (A, d)$, $\mathcal{O}_2(M_0, M_1, A) \rightarrow C$, $R(d, C) \rightarrow M_b$.

5.4 A Minimally Vulnerable Construction

We now describe a scheme that is minimally vulnerable to collusion. At the moment we do not address the usual Worry-Free Encryption properties, but will achieve a scheme which satisfies both the usual CPA security notions and the minimal vulnerability requirement shortly. The *Setup* phase will be as before, with the CA publishing a signing key *SignK* and storing a verification key VK for $\mathcal{S}$.

The pre-processing phase *Pre* will have the user run the first round of the oblivious transfer protocol $\mathcal{O}_1(x[i]) \rightarrow (\Sigma_i, d_i)$ so that the transfer bit on the i^{th} call is $x[i]$. That this is a parallel notion to our previous construction should now be clear, the user will only be granted access to one index at the i^{th} location. It then outputs $\Sigma = (\Sigma_i : i \in [1, n])$ and stores $SK = (d_i : i \in [1, n])$.

The Authorization phase *Auth* will simply consist of the central authority signing Σ, and the user's check phase *CheckAuth* will consist of checking this signature. Σ and the signature will be the user's public key. Note that the CA is not actually checking that the indices the user is receiving back in the OT protocol correspond to his credentials.

Encryption begins by checking the signature and making sure the public key is formatted as an n-tuple of 1st round messages from the OT protocol. Then, it returns $C_i = \mathcal{O}_2([f]_{i,b} : b \in [1, 2], \Sigma_i)$ the second round of the OT protocol on the two components of f at that index for $i \in [1, n]$. To decrypt run $R(d_i, C_i) \rightarrow [f]_{i,x[i]}$ for $i \in [1, n]$ and reconstruct $f(x)$. Call the above scheme $\mathcal{M}$.

THEOREM 5.1. *If $\mathcal{O}$ is a static 1-out-of-2 oblivious transfer scheme and (E, D) is a secure decomposable randomized encoding, $\mathcal{M}$ is minimally vulnerable to collusion.*

PROOF: For each index i, Σ_i has a bit associated, b_i such that any response of $\mathcal{O}_2$ using Σ_i as the first message will hide $[f]_{i,b}$ *information theoretically* by the static retrieval guarantee. Let *Ext* to be the extractor which finds such a bit for every index and labels the string of corresponding bits x', this is the string of bits corresponding to indices where the transfer leaks no information. Then, *Ext* outputs $x = \overline{x'}$. Note the challenge ciphertext can only depend on $[f]_{i,x[i]}$ for $i \in [1, n]$. The claim follows from the indistinguishability of the randomized encoding (recall privacy holds against non-uniform circuit families and therefore any σ_2, PK which allows A_3 a non-negligible probability to distinguish the encryptions of the encodings with non-negligible probably can be hard-wired into the circuit). □

A modification for Worry-Free Security. In order to achieve CPA Worry-Free Encryption guarantees along with the guarantee of minimal vulnerability, a slight modification will be needed. We will use a CPA secure Worry-Free Encryption scheme $\mathcal{W}$ and the above scheme $\mathcal{M}$.

Setup consists of the setup phases of both $\mathcal{W}$ and $\mathcal{M}$ running in parallel, the public parameters and MSK values of the new scheme will be the pair of corresponding outputs in

$\mathcal{W}$ and $\mathcal{M}$. Similarly, the *Auth* and *CheckAuth* phases will be the corresponding phases in $\mathcal{W}$ and $\mathcal{M}$ run on the corresponding input component. The only modification from both schemes running in parallel will be during encryption and decryption.

Encryption of a Pairwise-Independent Mask. Encryption takes as input $PK_{\mathcal{M}}, PP_{\mathcal{M}}, PK_{\mathcal{W}}, PP_{\mathcal{W}}$ along with a function $f : \{0,1\}^n \rightarrow \{0,1\}^k$. Let $\mathcal{H}_n^k$ be an efficiently sampleable family of pairwise independent hash functions from $\{0,1\}^n \rightarrow \{0,1\}^k$ of the same size s_n^k bounded by some polynomial in n and k.

Encryption will first sample $h \xleftarrow{\$} \mathcal{H}_n^k$ and generate $Enc(f + h, PK_{\mathcal{M}}, PP_{\mathcal{M}}), Enc(h, PK_{\mathcal{W}}, PP_{\mathcal{W}})$ and return pair of ciphertexts as the ciphertext[5]. Decrypt by decrypting $(f + h)(x)$ and $h(x)$ to recover $f(x)$. Call this scheme $\mathcal{Z}$.

THEOREM 5.2. *If $\mathcal{O}$ is a static 1-out-of-2 oblivious transfer scheme, (E, D) is a secure decomposable randomized encoding and $\mathcal{W}$ is a CPA Worry-Free Encryption Scheme, $\mathcal{Z}$ is a CPA secure Worry-Free Encryption Scheme minimally vulnerable to collusion.*

PROOF. Since the $\mathcal{M}$ component of the challenge ciphertext only depends on $([(f + h)]_{i,x[i]} : i \in [1, n])$ for the x that *Ext* outputs (defined identically to *Ext* for $\mathcal{M}$) and this is the only dependence of f, the scheme is *minimally vulnerable* identically to the proof of $\mathcal{M}$.

We now address *function hiding*. By the static retrieval guarantee of $\mathcal{O}$ and the indistinguishability property of E, the probability A_3 outputs a particular bit can only depend on $(f_z + h)(x')$ for one x' (found by *Ext*) from the first ciphertext component. Similarly, by the function hiding property of $\mathcal{W}$, the probability A_3 outputs b only depends on $h(x)$ for x the user's credentials from the second component. If $x = x'$, the adversary's probability only depends on $f_z(x) = f_{\bar{z}}(x)$ and otherwise, the two values are distributed uniformly at random by the pairwise independence of h. In either case, A_3's response can't depend on z by more than a negligible factor by the indistinguishability of (E, D).

Hidden credentials follows from the fact that the retrieved indices in the first key component are not revealed by the chooser security of $\mathcal{O}$ and not revealed in the second by the hidden credentials guarantee of $\mathcal{W}$.

Malicious Authority security follows since $PK_{\mathcal{M}}$ is generated completely by the user and the ciphertext component corresponding to $\mathcal{M}$ is the only one with dependence on the challenge function. Since both messages are computationally hidden from eavesdroppers in the OT protocol, this implies that the adversary can not distinguish the case where the challenge ciphertext component encrypted under $PK_{\mathcal{M}}$ is replaced by an encryption of a fixed string of the same length. Since the latter case has no dependence on the challenge function, this implies the adversary can only have negligible probability in guessing the challenge index. □

6. REFERENCES

[1] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *EUROCRYPT*, pages 119–135, 2001.

[2] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in nc^0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[3] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[4] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007.

[5] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *J. Comput. Syst. Sci.*, 60(3):592–629, 2000.

[6] Vipul Goyal. Reducing Trust in the PKG in Identity Based Cryptosystems. In *CRYPTO*, pages 430–447, 2007.

[7] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, pages 579–591, 2008.

[8] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[9] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *ICALP*, pages 244–256, 2002.

[10] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

[11] Jonathan Katz and Arkady Yerukhimovich. On black-box constructions of predicate encryption from trapdoor permutations. In *ASIACRYPT*, pages 197–213, 2009.

[12] Joe Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31, 1988.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *EUROCRYPT 2010*, pages 62-91.

[14] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In *CRYPTO*, pages 573–590, 1999.

[15] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *SODA*, pages 448–457, 2001.

[16] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.

[17] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[18] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In *TCC*, pages 457–473, 2009.

[19] Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In *ICALP (2)*, pages 560–578, 2008.

[20] A.C. Yao. Theory and application of trapdoor functions. In *FOCS*, pages 80–91, 1982.

[5] Note that we can assume $|f + h| = |g + h|$ if $|f| = |g|$ by using h as a black box in the circuit construction.