Towards Unified, Achievable, Simulation Based Function and Data Privacy in Functional Encryption

Shweta Agrawal * Abishek Kumarasubramanian † Amit Sahai ‡

Abstract

As the area of Functional Encryption (FE) has grown in importance, the landscape of its security definitions has become increasingly cluttered. Currently prevalent definitions are largely restricted to data privacy, and are broadly categorized as indistinguishability (IND) style or simulation (SIM) style, where the former are known to be insufficient and the latter are known to be unachievable even for simple functionalities like point functions (IBE) and inner products. Recently, Boneh et al. [BRS13a] introduced a new security definition to study the complementary aspect of function privacy in FE, but it is unclear how this definition relates to data privacy definitions.

In this work, we attempt to clean up this state of affairs and present a unified treatment of data and function hiding definitions. We propose a new simulation based definition for function privacy in addition to data privacy, which we call Relax-AD-SIM, and study its achievability. We show that Relax-AD-SIM interpolates simulation based (SIM) and indistinguishability based (IND) definitions for data privacy, and implies the function privacy definition of [BRS13a]. Our definition relaxes the requirements on the simulator to bypass impossibility of SIM in the standard model. We show that the inner product FE scheme of [LOS+10] enjoys Relax-AD-SIM security for data hiding while the weaker, selectively secure inner product FE scheme of [KSW08] suffices to achieve function hiding. Our definition provides the first unified simulation based definition that supports function hiding as well as data hiding. We emphasize that Relax-AD-SIM security is the strongest simulation based security definition known to be achievable by inner product FE to the best of our knowledge. This is because Relax-AD-SIM is an adaptive\textsuperscript{1} notion of simulation security, which, in its most general form (AD-SIM) is known to be impossible to achieve for inner product FE.

\*I.I.T, Delhi. Email: shweta.a@gmail.com.
\†UCLA Email: abishekk@cs.ucla.edu.
\‡UCLA Email: sahai@cs.ucla.edu.
\textsuperscript{1}i.e. it supports post challenge key queries
1 Introduction

In recent times, the area of Functional encryption (FE) [SW05, SW] has enjoyed immense importance in the field of cryptography. Functional encryption is a generalization of public key encryption which allows tremendous flexibility and control in learning information from encrypted data. In functional encryption, a user can be provided with a secret key corresponding to a function $f$, denoted by $SK_f$. Given $SK_f$ and ciphertext $CT_x = \text{Encrypt}(x)$, the user may run the decryption procedure to learn $f(x)$. Security of the system guarantees that nothing beyond $f(x)$ can be learned from $CT_x$ and $SK_f$.

Classic results in the area focused on constructing FE for restricted classes of functions. Identity based encryption (IBE) had already been constructed [Sha84, BF01, Coc01, BW06, GPV08,CHKP10, ABB10a,ABB10b] although it was not viewed as a special case of FE until later. Sahai and Waters provided the first construction for threshold functions [SW05] which was followed by rapid succession of ever-more powerful classes of functions: membership checking [BW07], boolean formulas [GPSW06, BSW07, LOS+10], inner product functions [KSW08, LOS+10, AFV11] and more recently, even regular languages [Wat12]. More contemporary constructions of Functional Encryption support general functions: Gurabov et al. [GVW13] and Garg et al. [GGH+13b] provided the first constructions for an important subclass of FE called “public index FE” (also known as “attribute based encryption”) for all circuits, Goldwasser et al. [GKP+13b] constructed succinct simulation-secure single-key FE scheme for all circuits. In a breakthrough result, Garg et al. [GGH+13a] constructed indistinguishability-secure multi-key FE schemes for all circuits. Goldwasser et al. and Ananth et al. [GKP+13a, ABG+13] also constructed FE for Turing machines. Recently, Functional Encryption has also been generalized to multi-input functional encryption [GGG+14].

Despite massive progress in constructing functional encryption systems for advanced classes of functions, it has remained unclear as to what is the “right” definition of security for FE systems. Security of FE comprises two orthogonal aspects: hiding the message or data embedded in the ciphertext, and hiding the function embedded in the key. A definitional framework studying data privacy for general FE was first proposed by Boneh, Sahai, and Waters [BSW11] and O’Neill [O’N10]. These and subsequent works elucidated the subtleties involved in defining a security model that captures meaningful real world data privacy. Despite considerable research [BSW11, O’N10, BO12, BF13, AGVW13, CIJ+13], what is the strongest, achievable notion for data privacy remains an actively debated open question. Unlike its counterpart, function privacy for FE has received comparatively less attention, primarily because some information about the function is inherently leaked by FE functionality in the public key setting. To see this, note that a holder of $SK_f$ can always encrypt messages $x_i$ of her choice and learn $f(x_i)$, thus learning information about the function. Recently, Boneh et. al. proposed a candidate security definition for function privacy [BRS13a] in the public key setting. In the symmetric key setting, function privacy was studied by Shen et al. [SSW09].

In this work, we revisit the security definitions for both data and function privacy of FE.

Data Privacy. Known definitions of data security for FE may be divided into two broad classes: Indistinguishability (IND) based or Simulation (SIM) based. Indistinguishability based security stipulates that it is infeasible to distinguish encryptions of any two messages, without getting a secret key that decrypts the ciphers to distinct values; simulation-based security stipulates that there exists an efficient simulator that can simulate the view of the adversary, given only the function evaluated on messages and keys. Boneh et al and O’Neill [BSW11, O’N10] showed that IND based definitions do not capture scenarios where it is required that the user learn only the output of the FE function, for e.g., when the function hides something computationally. To get around this, [BSW11, O’N10] proposed SIM based definitions that study FE in the “ideal world-real world” paradigm. However it has been shown that SIM based data security is impossible to achieve [BSW11, BO12], even when the adversary is restricted to pre-challenge queries [AGVW13]. This state of affairs raises the following question: Is there a relaxation\footnote{Non-adaptive SIM security has been considered where the attacker is only allowed to request keys before he sees the challenge. The focus of our work will be full adaptive SIM security} of SIM security for data privacy that is stronger than IND based security but bypasses impossibilities?
Function Privacy. In the symmetric key setting, function privacy was first studied by Shen et al. [SSW09], who define strong IND based security that captures both data and function privacy. In the public key setting, the first definition for function privacy was suggested very recently by Boneh et al. [BRS13a]. Their definition stems from the observation that if an adversary has some a-priori knowledge that the function comes from a small set, then he can encrypt messages that decrypt to different values for each candidate function, potentially leading to full recovery of the function embedded inside the key. They propose an IND style “real-or-random” definition of function privacy, which we call ReOrRand^{fn}, that stipulates that as long as the function $f$ was chosen from a sufficiently high min-entropy distribution, the adversary should not be able to distinguish $SK_f$ from a uniformly random secret key.

Boneh et al. [BRS13a] also suggest that function privacy for FE can be formalized in a framework inspired by program obfuscation. However, whereas obfuscation security is usually defined in simulation style, the definition in [BRS13a] is indistinguishability style. Therefore, the real-or-random definition for function privacy, while elegant and strong, does not compose well with the main accepted definition (SIM based) of data security. Thus, it is natural to ask: Is there a natural simulation based definition for function privacy in FE that is at least as strong as the real-or-random definition of [BRS13a]? Is such a definition achievable?

Our Results. In this work, we further explore meaningful definitions for FE security, and study relations with existing definitions. We present a new simulation based definition, which we call Relax-AD-SIM which captures data privacy as well as function privacy for FE. For data privacy, it interpolates simulation based (SIM) and indistinguishability based (IND) definitions, while for function privacy, it is at least as strong as the real-or-random definition of [BRS13a]. Our definition relaxes the requirement on the simulator to bypass impossibility of SIM based data security in the standard model. We show that the inner product FE scheme of [KSW08] enjoys Relax-AD-SIM security for function hiding and the inner product FE scheme of [LOS+10] enjoys Relax-AD-SIM security for data hiding as long as the predicate and data vectors are of constant dimension.

Related Work and Comparison. As stated above, the two existing notions of data privacy for FE are IND and SIM, both of which can be further classified as follows: [O’N10] described the divide between adaptive (AD) versus non-adaptive (NA) which captures whether the adversary’s queries to the key derivation oracle may or may not depend on the challenge ciphertext; and [GVW12] described the divide between one versus many, which depends on whether the adversary receives a single or multiple challenge ciphertexts. Thus, existing definitions of security belong to the class $\{1, \text{many}\} \times \{\text{NA, AD}\} \times \{\text{IND, SIM}\}$.

The study of simulation (SIM) based security for FE was initiated independently by Boneh et al. [BSW11] and O’Neill [O’N10]. Boneh, et al. [BSW11] showed an impossibility even for the “simple” IBE functionality under many-AD-SIM^{msg} security in the non programmable random oracle model. Bellare and O’Neill [BO12] put forward simulation-based definitions for Functional Encryption with non-black-box simulators. They also extended the lower bound for IBE [BSW11] to the setting of efficient, non-black-box simulators, assuming the existence of collision-resistant hash functions. Agrawal et al. [AGVW13] ruled out general functional encryption for 1-NA-SIM^{msg} security. Barbosa and Farshim [BF13] extended O’Neill’s equivalence between indistinguishability and semantic security to restricted adaptive key extraction attacks and show that this equivalence holds for a large class of functionalities. Intuitively, restricted adaptive simulation security restricts the key queries an adversary can make after he sees the challenge ciphertext to functions that are constant over the support of the message distribution, so as to circumvent the “non-committing encryption style” impossibility exhibited by [BSW11].

Achievability. While there has been fantastic progress for constructing FE schemes for advanced classes of functions secure under weaker definitions [GVW13, GGH+13b, GGH+13a], the situation is much less optimistic for achieving stronger definitions of security even for schemes supporting restricted classes of functions. The only positive results currently known for adaptive simulation based data privacy for FE are
by Boneh et al. [BSW11] for the IBE functionality in the Random Oracle Model. Note that the situation is better for non-adaptive simulation based data privacy – O’Neill [O’N10] showed that for certain classes of functions called preimage samplable functions, many-NA-IND$_{msg}$ and many-NA-SIM$_{msg}$ are equivalent. As mentioned above, Barbosa and Farshim [BF13] extended this equivalence to the restricted-adaptive setting, which is stronger than non-adaptive. However, even with this weakening of adaptivity, they are unable to achieve simulation based data privacy for inner product predicate FE; indeed they show that inner product functionality of [KSW08] can be used to encode a one way function under the Small Integer Solution (SIS) problem, and hence natural approaches to prove its restricted adaptive simulation security fail in the standard model. The problem of proving adaptive simulation based data privacy of inner product FE is explicitly left open both by [BSW11] and [BF13]. Recently, Caro et al. [CIJ+13] provided a “compiler” that converts IND based data security to SIM based data security for the Circuit-FE functionality.

The question of function privacy for FE has received even lesser attention. In the private key setting, the notion of function privacy was considered by Shen et al. [SSW09] who provided an indistinguishability style definition for data and function privacy. In the public key setting, Boneh et al [BRS13a] very recently provided the first security definition for function privacy of FE and achieved it for IBE [BRS13a] and subspace membership FE [BRS13b]. Their techniques however do not apply to the case of inner product FE and they leave open the question of achieving function privacy for the same. As described above, the function privacy definition of [BRS13a] is IND style, and does not compose well with SIM style data privacy. Our relaxation of the SIM based definition (which we call Relax-AD-SIM) is inspired from “weak simulation” definitions in the obfuscation literature [Wee05, CRV10], which allow the simulator to have an inverse polynomial dependency on the distinguishing probability and also permit the simulator to accept non black box adversary specific advice.

Bypassing Impossibility. We briefly examine how our techniques can be used to bypass the impossibility of many-AD-SIM$_{msg}$ secure inner product FE [BSW11]. Recall the main idea behind the impossibility of many-AD-SIM$_{msg}$ secure IBE in the standard model provided by [BSW11]. At a high level, the argument forced the simulator to commit to a challenge ciphertext before seeing the adaptive (or more precisely post-challenge) key queries of the adversary. Thus, the challenge ciphertext could not have been programmed to satisfy the requisite relations with the post-challenge key queries. By choosing the challenge CT to encrypt random bits and setting the number of challenge ciphertexts to be greater than the length of the secret key, the successful simulator is forced to achieve an information theoretic compression of random bits, which is impossible. For more details we refer the reader to [BSW11].

Our definition for relaxed simulation Relax-AD-SIM$_{msg}$ bypasses the above problem by allowing the simulator to make more queries than the adversary. For the case of data hiding, the simulator is allowed to make not just the queries that the adversary makes in the current run of the experiment, but also all the extra queries that the adversary could have made with noticeable probability, i.e. queries it might make for a different outcome of its random coin flips. Due to this, the simulator can program the ciphertext not just for queries asked by the adversary so far, but also for queries that the adversary is likely to ask in the future. While this seems like a very weak definition at first, we show, surprisingly, that it implies the well accepted AD-IND$_{msg}$ definition for data privacy. This is because any successful AD-IND$_{msg}$ adversary can be converted to a self censoring Relax-AD-SIM$_{msg}$ adversary, who for any sequence of random coins, can only make queries with nonzero probability that do not allow distinguishing between the IND challenge messages. Thus, the simulator is allowed to make extra queries to the oracle, but none that allow it to distinguish between the challenge messages. The formal proof is provided in Section 2.

We note that our definition does not bypass the 1-NA-SIM$_{msg}$ impossibility exhibited by Agrawal et al. [AGVW13]. Indeed, we show that our definition is strictly stronger than 1-NA-SIM$_{msg}$ (and strictly weaker than 1-AD-SIM$_{msg}$).

Organization of Paper. In Appendix A, we introduce functional encryption and provide known definitions for data and function privacy. In Section 2, we introduce our new notion of relaxed adaptive
simulation based security for both data and function hiding. In Section 3, we analyze its relationship with known definitions of security for both data and function hiding. In Section 4 we recap known constructions of inner product FE for completeness. In Section 5, we show that the inner product FE construction achieves both data and function hiding according to our relaxed adaptive simulation definition, for both data and function privacy. We conclude in Section 6.

2 Relaxed Simulation Security for Functional Encryption

In light of the strong impossibility results surrounding simulation based security definitions, we propose a relaxation of simulation security which is achievable in the standard model, for both data and function privacy. Although our relaxation seems counter-intuitive at first, we will show that this relaxation is meaningful and implies the IND based definition of data hiding (Section A.2) as well as real-or-random definition for function hiding [BRS13a] (Section A.3).

2.1 Definition of Data Hiding

The simulation based definition of data hiding is similar to the one presented in Section A.2 except that we relax the definition of admissible simulator. We allow the size of the simulator to have an inverse polynomial dependency on the distinguishing probability. We also allow the simulator to make more queries than the adversary, as long as it restricts itself to queries made by the adversary with noticeable probability. More details follow. For completeness, we recap the experiments here.

Definition 2.1 (Real and Ideal experiments.). Let $\mathcal{FE}$ be a functional encryption scheme for a circuit family $\mathcal{C}$. Consider a p.p.t. adversary $A = (A_1, A_2)$ and a stateful p.p.t. simulator $S$. Let $U_x(\cdot)$ denote a universal oracle, such that $U_x(C) = C(x)$. Consider the following two experiments:

<table>
<thead>
<tr>
<th>$\text{exp}_{\mathcal{FE}, A}^{\text{real msg}}(1^k)$</th>
<th>$\text{exp}_{\mathcal{FE}, S}^{\text{ideal msg}}(1^k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $(PK, MSK) \leftarrow \text{Setup}(1^k)$</td>
<td>1: $PK \leftarrow S_1(1^k)$</td>
</tr>
<tr>
<td>2: $(\vec{x}, st) \leftarrow A_1^{\text{KeyGen}(MSK, \cdot)}(PK)$</td>
<td>2: $(\vec{x}, st) \leftarrow A_1^{S_1(\cdot)}(PK)$</td>
</tr>
<tr>
<td>3: $\vec{C}T \leftarrow \text{Encrypt}(PK, \vec{x})$</td>
<td>3: $\vec{C}T \leftarrow S_2^{U_x(\cdot)}(1^k, 1</td>
</tr>
<tr>
<td>4: $\alpha \leftarrow A_2^{O(\cdot)(MSK, \cdot)}(PK, \vec{C}T, st)$</td>
<td>4: $\alpha \leftarrow A_2^{S_2(\cdot)}(PK, \vec{C}T, st)$</td>
</tr>
<tr>
<td>5: Output $(\vec{x}, \alpha)$</td>
<td>5: Output $(\vec{x}, \alpha)$</td>
</tr>
</tbody>
</table>

The experiments $\text{exp}_{\mathcal{FE}, A}^{\text{real msg}}(1^k)$ and $\text{exp}_{\mathcal{FE}, S}^{\text{ideal msg}}(1^k)$ are defined exactly as in the standard simulation based definition (Please see Appendix 2.1). To define our admissible simulator, we will need some notation. Note that the real world adversaries $A_1, A_2$ are randomized, and can make different queries in different runs of the experiment. Also note that $A_2$’s queries may depend on $A_1$’s queries; in particular, $A_2$ may be restricted from making certain queries that leak information about $x$. Such restrictions are encoded in the state variable $st$ that is passed from $A_1$ to $A_2$. Let $Q_1$ be the set of queries made by adversary $A_1$ in a run of the experiment $\text{exp}_{\mathcal{FE}, S}^{\text{ideal msg}}(1^k)$. Let $Q_2$ be the set of queries that are likely to be made by adversary $A_2$, i.e. the queries that are made by $A_2$ with probability greater than $\epsilon$, where probability is taken over the random coins of $A_2$, conditioned on $st$.

Definition 2.2 (Admissible simulator). For fixed $\epsilon > 0$, an admissible Relax-AD-SIM$^{\text{msg}}$ simulator $S_\epsilon = (S_1, S_2)$ is such that:

- $S_1$ runs in time $\text{poly}(k)$.

\(^3\)One can replace a stateful simulator by a regular (stateless) simulator that outputs a state $st_\epsilon$ upon each invocation which is carried over to its next invocation.
\begin{itemize}
  \item $S_2$ runs in time \(\text{poly}(\kappa, 1/\epsilon)\), may depend on adversary \(A_2(\text{st})\) where \(\text{st}\) is output by \(A_1\), and makes queries from the set \(Q_S\) where \(|Q_S| = \text{poly}(\kappa, 1/\epsilon)\) and \(Q_S \subseteq Q_1 \cup Q_2\). Note that \(Q_1\) and \(Q_2\) are collectively encoded in \(\text{st}\) and description of \(A_2\).
\end{itemize}

It is important to note in particular that if the adversary \(A_2\) makes some query with probability less than \(\epsilon\), then the admissible simulator \(S_2\) is disallowed from making that query.

**Definition 2.3.** The functional encryption scheme \(FE\) is then said to be Relax-\(\text{AD-SIM}^{\text{msg}}\) secure, if for every p.p.t. adversary \(A_1\), there exists an admissible simulator \(S_1\) and for all \(A_2\) and state \(\text{st}\) output by \(A_1\) there exists an admissible simulator \(S_2\) such that for all p.p.t. distinguishers \(D\),

\[\left| \Pr \left\{ D \left[ \exp^{\text{real msg}}_{FE,A}(1^\kappa) \right] \Rightarrow 1 \right\} - \Pr \left\{ D \left[ \exp^{\text{ideal msg}}_{FE,S}(1^\kappa) \right] \Rightarrow 1 \right\} \right| \leq \epsilon\]

### 2.2 Definition of Function Hiding

We provide a similar simulation based definition for function hiding. The goal of function hiding is to quantify the amount of information leaked by \(SK_C\) about the circuit \(C\). The strongest possible goal in this aspect would be to demand that the algorithm \(\text{KeyGen}\) essentially provide an obfuscator for the circuit family \(C_\kappa\). Due to known impossibility results for obfuscation we propose our approximate notion of simulation identical in spirit to related works on obfuscation [Wee05, CRV10].

**Definition 2.4** (Real and Ideal experiments for function hiding). Let \(FE\) be a functional encryption scheme for a circuit family \(C\). Consider a p.p.t. adversary \(A = (A_1, A_2)\) and a stateful p.p.t. simulator \(S\). Let \(W_C(\cdot)\) denote a universal oracle, such that \(W_C(x) = C(x)\). Consider the following two experiments:

<table>
<thead>
<tr>
<th>exp^{\text{real fn}}_{FE,A}(1^\kappa):</th>
<th>exp^{\text{ideal fn}}_{FE,S}(1^\kappa):</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: ((PK, MSK) \leftarrow \text{Setup}(1^\kappa))</td>
<td>1: (PK \leftarrow S(1^\kappa))</td>
</tr>
<tr>
<td>2: ((C, \text{st}) \leftarrow A_1^{\text{KeyGen}(MSK,:)}(PK))</td>
<td>2: ((C, \text{st}) \leftarrow A_1^S(\cdot)(PK))</td>
</tr>
<tr>
<td>3: (SK_C \leftarrow \text{KeyGen}(PK, C))</td>
<td>3: (SK_C \leftarrow S_{W_C(\cdot)}(1^\kappa, 1^{</td>
</tr>
<tr>
<td>4: (\alpha \leftarrow A_2^{O(MSK,:)}(PK, SK_C, \text{st}))</td>
<td>4: (\alpha \leftarrow A_2^S(\cdot)(PK, SK_C, \text{st}))</td>
</tr>
<tr>
<td>5: Output (\alpha)</td>
<td>5: Output (\alpha)</td>
</tr>
</tbody>
</table>

**Definition 2.5** (Admissible simulator). For any \(\epsilon > 0\), the admissible Relax-\(\text{AD-SIM}^{\text{Fn}}\) simulator \(S_2\) for function hiding runs in time \(\text{poly}(\kappa, 1/\epsilon)\), makes \(\text{poly}(\kappa, 1/\epsilon)\) queries to the function oracle \(W_C\) (without any restriction on the nature of queries) and may accept non black box advice about adversary \(A_2\).

Note that unlike the case of data hiding, the queries of the admissible function hiding simulator are unrestricted. This is because oracle queries in this case correspond to encryption queries not secret key queries: since any adversary given \(SK_C\) can encrypt messages \(x_i\) of her choice and run the decrypt algorithm to learn \(C(x_i)\), the corresponding simulator must be allowed to request function values \(C(x_i)\) for any \(x_i\) of its choice. As in the case of data hiding, Relax-\(\text{AD-SIM}^{\text{Fn}}\) security can be classified into the following two types:

1. **The adaptive case** which allows both pre-challenge and post-challenge queries, i.e. where:
   - the oracle \(O(MSK, \cdot) = \text{KeyGen}(MSK, \cdot)\) and
   - the oracle \(O'(\cdot)\) is the simulator, namely \(S_{W_C(\cdot)}(\cdot)\)

2. **The non-adaptive case**, which only allows pre-challenge queries, i.e. where the oracles \(O(MSK, \cdot)\) and \(O'(\cdot)\) are both the “empty oracles” that return nothing.
Definition 2.6. The functional encryption scheme \( \mathcal{FE} \) is then said to be Relax-AD-SIM\(^{\text{Fn}} \) secure, if for every p.p.t. adversary \( A = (A_1, A_2) \), every \( \epsilon > 0 \), there exists an admissible stateful p.p.t. simulator \( S_\epsilon \) such that for all p.p.t distinguishers \( D \),

\[
| \Pr \{ D[\exp_{\mathcal{FE},A}(1^n)] \Rightarrow 1 \} - \Pr \{ D[\exp_{\mathcal{FE},S}(1^n)] \Rightarrow 1 \} | \leq \epsilon
\]

2.3 What does Relax-AD-SIM mean?

Our relaxed simulation based definition Relax-AD-SIM for data and function privacy is somewhat counter-intuitive. Allowing the simulator to make extra queries seems to imply a very weak notion of security. However, we show that it is stronger than the well accepted notions of security for data as well as function hiding. Thus, it sheds light on the meaning of adaptive IND based data privacy as well as real-or-random function privacy. We do not claim that Relax-AD-SIM is the right notion of simulation based security, but we do believe it is an important step in that direction. Our motivation for considering this definition was twofold – for data privacy, we wished to find a weakening of adaptive simulation based security that is stronger than adaptive IND based security (AD-IND\(^{\text{msg}} \)), but nevertheless bypasses the impossibilities exhibited by [BSW11, AGVW13]. For function privacy we wished to find a simulation based definition analogous to the accepted SIM based definition of data privacy, which nonetheless implies the real-or-random function hiding definition of [BRS13a].

In Section 3.1, we establish that Relax-AD-SIM\(^{\text{msg}} \) implies the widely accepted AD-IND\(^{\text{msg}} \). We also show that Relax-1-AD-SIM\(^{\text{msg}} \) implies 1-NA-SIM\(^{\text{msg}} \). In Section 3.2, we establish that Relax-AD-SIM\(^{\text{Fn}} \) implies the Real-or-Random notion of function privacy defined by [BRS13a].

3 Connections to existing simulation based definitions

In this section we explore the connections of Relax-AD-SIM to existing definitions for both data and function hiding in functional encryption.

3.1 Data Hiding

We establish that Relax-AD-SIM\(^{\text{msg}} \) implies the widely accepted AD-IND\(^{\text{msg}} \). We also show that Relax-1-AD-SIM\(^{\text{msg}} \) implies 1-NA-SIM\(^{\text{msg}} \).

Claim 3.1. Relax-AD-SIM\(^{\text{msg}} \) \( \Rightarrow \) AD-IND\(^{\text{msg}} \)

Proof. We will show that \( \neg \) AD-IND\(^{\text{msg}} \) \( \Rightarrow \) \( \neg \) Relax-AD-SIM. Given a successful AD-IND\(^{\text{msg}} \) adversary \( (A_1, A_2) \) (refer Section A.2 for definition of AD-IND\(^{\text{msg}} \)), we will construct a real world Relax-AD-SIM adversary \( (B_1, B_2) \) as follows: \( B_1 \) runs \( A_1 \), receives \( (x_0, x_1) \), flips a bit \( b \) and outputs CT\((x_b)\) as the challenge CT. Now \( B_2 \) receives CT\((x_b)\) and runs \( A_2 \) to learn \( b \). It answers \( A_2 \)'s key queries using its keygen oracle.

\( B_2 \) makes exactly the same key queries as \( A_2 \) in order to simulate it. In particular,

\[
\Pr (B_2 \text{ queries } v^* \text{ s.t. } f(x_0, v^*) \neq f(x_1, v^*)) = 0
\]

since an admissible \( A_2 \) cannot make such a query.

Now consider an admissible simulator \( S_\epsilon \) who must simulate \( (B_1, B_2) \). \( S_\epsilon \) has oracle access to \( f(x_b, \cdot) \) and can query the oracle for any key \( v^* \) which could have been queried by \( (B_1, B_2) \) with probability \( \epsilon \). Since \( \Pr (B_2 \text{ queries } v^* \text{ s.t. } f(x_0, v^*) \neq f(x_1, v^*)) = 0 \), \( S_\epsilon \) cannot make queries that distinguish \( x_0 \) from \( x_1 \). Thus, the simulator does not learn anything about \( b \) from the function oracle, hence cannot simulate CT\((x_b)\) with non-negligible advantage.

We also show that Relax-1-AD-SIM\(^{\text{msg}} \) implies 1-NA-SIM\(^{\text{msg}} \).

Claim 3.2. Relax-1-AD-SIM\(^{\text{msg}} \) \( \Rightarrow \) 1-NA-SIM\(^{\text{msg}} \)
Proof. Again, we show that $\neg 1$-NA-SIM$^{msg}$ $\Rightarrow$ $\neg$ Relax-1-AD-SIM$^{msg}$. Thus, we convert a successful 1-NA-SIM$^{msg}$ adversary into a successful Relax-1-AD-SIM$^{msg}$ adversary. The Relax-1-AD-SIM$^{msg}$ adversary is exactly equal to the 1-NA-SIM$^{msg}$ adversary. The simulator in the Relax-1-AD-SIM$^{msg}$ experiment is more powerful than in the 1-NA-SIM$^{msg}$ case, and can make queries to the function oracle that $A_2$ makes with probability at least $\epsilon$. However, the 1-NA-SIM$^{msg}$ adversary $A_2$ makes no key queries by definition, hence probability that it queries any value is 0. Hence, the Relax-1-AD-SIM$^{msg}$ simulator does not have any power over the 1-NA-SIM$^{msg}$ simulator, and cannot simulate the Relax-1-AD-SIM$^{msg}$ adversary.

\[\square\]

3.2 Function Hiding

In this section, we show that Relax-AD-SIM$^{Fn}$ implies real or random function hiding as defined by [BRS13a]. At a high level, the function privacy definition of [BRS13a, BRS13b] stipulates that as long as a circuit $C_{real}$ is chosen with “sufficient” unpredictability, an adversary should not be able to tell the difference between $SK(C_{real})$ and $SK(C_{rand})$. Their notion of sufficiently unpredictable, depends on the underlying circuit family. We begin by defining the notion of feasible entropy distributions which abstracts the “sufficient unpredictability” property required by [BRS13a, BRS13b]. A distribution $D$ is called a feasible entropy distribution, if a circuit sampled from $D$ cannot be differentiated from a uniformly sampled circuit, given just oracle access to the circuit. We show that Relax-AD-SIM$^{Fn}$ $\Rightarrow$ ReOrRand$^{Fn}$ for any FE scheme where functions corresponding to the secret keys are sampled from a feasible entropy distribution $D$.

Definition 3.3 (Feasible Entropy Distributions). Let $W_C(\cdot)$ denote a universal functional oracle defined as $W_C(x) = C(x)$, let $C_n$ denote a family of circuits and let $D$ any distribution on $C_n$. Then, we say that $D$ is a feasible entropy distribution, if for all non-uniform polynomial time algorithms $S$, it holds that:

\[
\left| \Pr_{C \in C_n} (S^{W_C(1^n)} \neq 1) - \Pr_{C \in C_n} (S^{1^n} \neq 1) \right| \leq \text{negl}(\kappa)
\]

We define feasible entropy distribution for a single challenge function but this can be generalized to multiple challenge functions, as in [BRS13a]. We observe that the input distributions defined for the IBE functionality and the subspace product (or inner product) functionality defined by [BRS13a, BRS13b] satisfy Definition 3.3.

Claim 3.4. Relax-AD-SIM$^{Fn}$ $\Rightarrow$ ReOrRand$^{Fn}$ \footnote{see Definitions A.10 and 2.4} for any functional encryption scheme where the functions corresponding to the secret keys are sampled from a feasible entropy distribution $D$.

Proof. Assume that there is an admissible ReOrRand$^{Fn}$ adversary $A = (A_1, A_2)$ who distinguishes between real and random with non-negligible advantage $\alpha$ (see Definition 2.4). We will build a real world Relax-AD-SIM$^{Fn}$ adversary $(B_1, B_2)$ as follows: $B_1$ runs $A_1$ and receives $D$. $B_1$ samples a circuit $C_{real}$ from $D$ and a circuit $C_{rand}$ uniformly. Next, it flips a bit $b \in \{\text{real, rand}\}$ and outputs $C_b$. Now $B_2$ receives $SK(C_b)$, runs $A_2$, learns $b$ and outputs $b$ as his view. We assume that $(B_1, B_2)$ can be simulated by admissible simulator $S$, where $\epsilon$ is a non-negligible quantity to be set later and arrive at a contradiction.

Consider the following four hybrids.

1. In this game, $B_2$ is given the output of the honest keygen algorithm on a circuit $C_{real}$ drawn from the feasible entropy distribution, i.e. $SK_{C_{real}} = \text{KeyGen}(C_{real})$. $B_2$ outputs some View$_1$.

2. In this game, we replace the challenge secret key to be the output of ideal world simulator $S$ who, given access to the function oracle $W_{C_{real}}(\cdot)$, runs in time poly($\kappa, 1/\epsilon$) and outputs some $SK_{C_{real}}$. Now $B_2$ outputs some View$_2$.

3. In this game we replace the function oracle of $S$ with $W_{C_{rand}}(\cdot)$, the rest of the game remains the same. Say that $B_2$ outputs some View$_3$. 
In this section, we will show that this construction achieves Adaptive Security, namely $\alpha$. Since $\epsilon$ can be arbitrary, by setting $\epsilon = \frac{\alpha}{100}$ we then arrive at a contradiction.

## 4 Construction of Inner Product FE

In this section, we recap the construction for Inner Product FE from [KSW08] for completeness. In the following section, we will show that this construction achieves Relax-AD-SIM security for function hiding. For data hiding, we will need an inner product FE scheme that achieves adaptive security, namely [LOS+10] but for ease of exposition we describe the simplest possible scheme here. We stress that we do not provide a new construction. The reader who is familiar with [KSW08] may skip ahead to the following section.

To begin we define some notation that will be useful to us.

**Notation for Linear Algebra over groups.** When working over the prime order group $G$, we will find it convenient to consider tuples of group elements. Let $\vec{v} = (v_1, \ldots, v_d) \in \mathbb{Z}_p^d$ for some $d \in \mathbb{Z}$ and $g \in G$. Then we define $g^\vec{v} = (g^{v_1}, \ldots, g^{v_d})$. For ease of notation, we will refer to $(g^{v_1}, \ldots, g^{v_d})$ by $(v_1, \ldots, v_d)$. This notation allows us to do scalar multiplication and vector addition over tuples of group elements as:

$$(g^\vec{v})^a = g^{(a\vec{v})} \text{ and } g^\vec{v} \cdot g^\vec{w} = g^{(\vec{v} + \vec{w})}$$

Finally we define a new function, $\vec{e}$ which deals with pairings two $d$-tuples of elements $\vec{v}, \vec{w}$ as:

$$\vec{e}(g^\vec{v}, g^\vec{w}) := \prod_{i=1}^{d} e(g^{v_i}, g^{w_i}) = e(g,g)^{\vec{v} \cdot \vec{w}}$$

where the vector dot product $\vec{v} \cdot \vec{w}$ in the last term is taken modulo $p$. We represent an element $g^a \in G$ using the notation $(a)$ and an element $e(g,g)^b \in G_T$ using the notation $[b]$. Here $g$ is assumed to be some fixed generator of $G$.

**Dual Pairing Vector Spaces.** We will employ the concept of dual pairing vector spaces from [Lew12, OT08, OT09]. For a fixed dimension $d$, Let $B = (b_1, \ldots, b_d), B^* = (b_1^*, \ldots, b_d^*)$ be two random bases (represented as column vectors) for the vector space $\mathbb{Z}_p^d$. Furthermore, they are chosen so that,

$$
\begin{pmatrix}
\vec{b}_1 \\
\vdots \\
\vec{b}_d
\end{pmatrix} \cdot 
\begin{pmatrix}
\vec{b}_1^* \\
\vdots \\
\vec{b}_d^*
\end{pmatrix}
= \psi \cdot I_{d \times d}
$$

(1)

where $I_{d \times d}$ is the identity matrix and $\psi \overset{\$}{\leftarrow} \mathbb{G}_p$ is a uniformly distributed random variable. [Lew12] describes a standard procedure which allows one to pick such bases.

We use the notation $(B, B^*) \overset{\$}{\leftarrow} Dual(\mathbb{Z}_p^d)$ in the rest of this work to describe the selection of such basis vectors. Depending on the scheme we choose the value of $d$. We now establish further notation using

---

4. In this game, $B_2$ is given the output of the honest keygen algorithm on a circuit $C_{\text{rand}}$ drawn from the uniform distribution, i.e. $SK_{C_{\text{rand}}} = \text{KeyGen}(C_{\text{rand}})$. $B_2$ outputs some $\text{View}_4$. We merely observe that hybrids 1, 2 are at most $\epsilon$ distance away because of the scheme $\mathcal{FE}$ satisfies Definition 2.4. Hybrids 2, 3 are negligibly close to each other because of the fact that the distribution $\mathcal{D}$ over the circuit family $C_{\kappa}$ satisfies the property defined in Definition 3.3. Hybrid 3, 4 are at most $\epsilon$ distance away by Definition 2.4.

Thus we have that for any $\epsilon$, Hybrid 1 and Hybrid 4 are separated by at most $2\epsilon$ (plus some negligible quantity) distance. However since $A = (A_1, A_2)$ guesses $b$ correctly with non-negligible advantage $\alpha$, by definition we have that the distance between Hybrid 1 and Hybrid 4 is $> \alpha$. Since $\epsilon$ can be arbitrary, by setting $\epsilon = \frac{\alpha}{100}$ we then arrive at a contradiction. 

---

5. Here, note that in the vein of Section 4, we are referring to vectors of group elements.
three tuple of formal polynomials \((a, b, c)\) both achieve relaxed simulation security (Section 2) for inner product FE presented in [KSW08]. It then applies a series of transformations, as developed in [GKSW10, Fre10, OT08, OT09, Lew12], to convert it to a scheme over prime order groups. We show that it achieves relaxed simulation security (Section 2) for both data and function privacy.

The functionality \(\mathcal{F} : \mathbb{Z}_p^n \times \mathbb{Z}_p^n \rightarrow \{0, 1\}\) is described as \(\mathcal{F}(\vec{x}, \vec{v}) = 1\) iff \((\vec{x} \cdot \vec{v})\) is even. Let \(\text{GroupGen}\) be a group generation algorithm which takes as input a security parameter \(\kappa\) and outputs a bilinear group of prime order \(p\) with length \(p = \kappa\).

- **Setup(\(\kappa\)):** Let \((p, G, G_T, e) = \text{GroupGen}(\kappa)\). Let \(n \in \mathbb{Z}, n > 1\) be the dimension of the message space. Pick \((\mathbb{B}, \mathbb{B}^\ast) \xleftarrow{\$} \text{Dual}(\mathbb{Z}_p^3)\) and let \(P, Q, R, R_0, H_1, H_2, H_3, R_2, \ldots, H_n, R_n \xleftarrow{\$} \mathbb{Z}_p\). Set,

  \[
  \begin{align*}
  \text{PK} & = (p, G, G_T, e) \\
  \text{EK} & = (P \cdot \vec{b}_1, Q \cdot \vec{b}_2 + R_0 \cdot \vec{b}_3, R \cdot \vec{b}_3, \{H_i \cdot \vec{b}_1 + R_i \cdot \vec{b}_3\}_{i=1}^{i=n}) \\
  \text{MSK} & = (Q, \{H_i\}_{i=1}^{i=n}, \vec{b}_1, \vec{b}_2, \vec{b}_3, \vec{b}_1^\ast, \vec{b}_2^\ast, \vec{b}_3^\ast)
  \end{align*}
  \]

- **Encrypt(\(\text{EK}, \vec{x}\)):** Let \(\vec{x} = (x_1, \ldots, x_n), x_i \in \mathbb{Z}_p\). Choose random \(s, \alpha \in \mathbb{Z}_p\) and random elements \(\{r_i\}_{i=1}^{i=n} \subseteq \mathbb{Z}_p\).

  \[
  \text{CT}\_x = \left\{ C_0 = sP \cdot \vec{b}_1, C_i = \left\{ s(H_i \cdot \vec{b}_1 + R_i \cdot \vec{b}_3) + \alpha \cdot x_i \cdot (Q \cdot \vec{b}_2 + R_0 \cdot \vec{b}_3) + r_i \cdot \vec{b}_3 \right\}_{i=1}^{i=n} \right\}
  \]

- **KeyGen(\(\text{MSK}, \vec{v}\)):** Let \(\vec{v} = (v_1, \ldots, v_n), v_i \in \mathbb{Z}_p\). Choose, \(\{\delta_i\}_{i=1}^{i=n}, \zeta, Q_6, R_5 \xleftarrow{\$} \mathbb{Z}_p\) and construct \(\text{SK}_\vec{v} = (K_0, K_1, \ldots, K_n)\) as,

  \[
  K_0 = \left( -\sum_{i=1}^{n} H_i \cdot \delta_i \right) \cdot \vec{b}_1^\ast + Q_6 \cdot \vec{b}_2^\ast + R_5 \cdot \vec{b}_3^\ast
  \]

  and,

  \[
  \left\{ K_i = \delta_i \cdot P \cdot \vec{b}_1^\ast + Q \cdot \zeta \cdot v_i \cdot \vec{b}_2^\ast \right\}_{i=1}^{i=n}
  \]
• Decrypt(SK, CT): Compute \( b = e(C_0, K_0) \cdot \prod_{i=1}^{i=n} e(C_i, K_i) \) and output 1 if \( b = e(g, g)^0 \) and 0 otherwise.

Intentionally leaked information as defined in Remark 2 for the above scheme is \( n \), the length of the message and key space. Correctness of the scheme relies on the cancellation properties between the vectors in \( \mathbb{B} \) and \( \mathbb{B}^* \) as described in Eqn 1. We provide proof of correctness in Appendix B.

5 Achieving relaxed simulation security

In this section, we provide a proof that the inner product FE scheme presented in Section 4.1 satisfies the Relax-AD-SIM definition of data and function hiding (Definitions 2.3, 2.4). We begin with the case of function hiding.

5.1 Function Hiding

Our proof directly uses the construction and proof of security of the inner product obfuscation scheme presented by Canetti et al. [CRV10].

CRV Obfuscation of Hyperplane Membership: Canetti et al. [CRV10] construct an obfuscator CRV-Obf for the inner product functionality under the following assumption: given a tuple of group elements \( (g^{a_1}, g^{a_2}, \ldots, g^{a_d}) \) where the \( a_i \)'s are chosen from some joint distribution, \( g^{p(a_1, \ldots, a_d)} \) is indistinguishable from uniform for all polynomials \( p \) except those which are linear when restricted to the support of the joint distribution. For the precise formulation, we refer the reader to [CRV10], Assumption 5.

**Theorem 5.1.** [CRV10] Let \( \vec{v} \in \mathbb{Z}_p^n \) and CRV-Obf\( _{\vec{v}} \) be the obfuscator for \( \vec{v} \) constructed by Canetti et al. [CRV10]. Then, for fixed \( \epsilon > 0 \), there exists a simulator \( S'_{\text{ad}} \) who takes non black box advice \( V \) about adversary \( A \), runs in time \( \text{poly}(\kappa, 1/\epsilon) \) and outputs a “dummy” obfuscation CRV-Obf’\( _{\vec{v}} \) so that

\[
| \Pr (A(\text{CRV-Obf}_{\vec{v}}) \Rightarrow 1) - \Pr (A(\text{CRV-Obf}'_{\vec{v}}) \Rightarrow 1) | \leq \epsilon
\]

Consider the relaxed simulation based definition of function hiding (Relax-AD-SIM\( ^{\mathbb{F}_n} \)). Let CRV-Obf\( _C \) be the obfuscation of \( C \) as output by the scheme of [CRV10]. We present our proof in two steps. First, we define a hybrid world which is identical to the real world except that the output of the KeyGen algorithm is replaced by the output of adversary \( A_2 \) given CRV-Obf\( _C \) as input instead of \( C \). We claim that the algorithm KeyGen of Section 4.1 can be rewritten in a way that it produces the correct output when given input CRV-Obf\( _C \) instead of \( C \). Hence, the real world is indistinguishable from the hybrid world. The transition from the hybrid world to the ideal world is accomplished by invoking the proof of security of the CRV-obfuscator [CRV10].

First we show how the algorithm KeyGen of Section 4.1 can be rewritten to produce the correct output when given input CRV-Obf\( _C \) instead of \( C \). To see this, recall from Section 4.1, that the keygen algorithm KeyGen is,

\[
\text{KeyGen(MSK, } \vec{v}) \text{: Let } \vec{v} = (v_1, \ldots, v_n), v_i \in \mathbb{Z}_p. \text{ Choose, } \{ \delta_i \}_{i=1}^{i=n}, \zeta, Q_6, R_5 \overset{\$}{\leftarrow} \mathbb{Z}_p \text{ and construct } \text{SK}_{\vec{v}} = (K_0, K_1, \ldots, K_n) \text{ as,}
\]

\[
K_0 = \left( - \sum_{i=1}^{i=n} H_i \cdot \delta_i \right) \cdot \vec{b}^* + Q_6 \cdot \vec{b}_2^* + R_5 \cdot \vec{b}_3^*
\]

and,

\[
\{ K_i = \delta_i \cdot P \cdot \vec{b}_1^* + Q \cdot \zeta \cdot v_i \cdot \vec{b}_2^* \}_{i=1}^{i=n}
\]

We would also like to observe that the construction of the obfuscator CRV-Obf in our notation is,

\[
\text{CRV-Obf}(\vec{v}) \text{: Let } \vec{v} = (v_1, \ldots, v_n), v_i \in \mathbb{Z}_p. \text{ Choose } \zeta \overset{\$}{\leftarrow} \mathbb{Z}_p \text{ and construct } \text{CRV-Obf}_{\vec{v}} = (\zeta \cdot v_1, \ldots, \zeta \cdot v_n).
\]
A_2 upon receiving n components from CRV-Obf_C directly plugs these into the construction KeyGen as described above and computes SK_C that is identical to the real world.

The transition from the hybrid world to the ideal world is accomplished by invoking the proof of security of the CRV-obfuscator [CRV10]. By Theorem 5.1, for fixed \( \epsilon > 0 \) and obfuscator CRV-Obf_C, Canetti et al. [CRV10] construct a simulator \( S'_e \) which accepts non black box advice \( V \) about adversary \( A_2 \), runs in time \( \text{poly}(\kappa, 1/\epsilon) \) and makes \( \text{poly}(\kappa, 1/\epsilon) \) oracle queries for points in the set \( V \) and outputs a “dummy” obfuscator CRV-Obf'_C. This dummy obfuscator is indistinguishable by adversary \( A_2 \) from the real obfuscator CRV-Obf_C except with probability \( \epsilon \).

Formally, algorithm \( S_e \), given non black box advice \( V \) about adversary \( A_2 \) works as follows:

1. \( S_e \) runs the honest Setup algorithm ensuring that the prime order group \( G \) used is identical to the one instantiated by the obfuscation algorithm CRV-Obf.
2. \( S_e \) uses the honest KeyGen algorithm to answer key requests of adversary \( A_1 \).
3. Run the CRV simulator \( S'_e(V) \) and obtain the dummy obfuscator CRV-Obf'_C.
4. Run the KeyGen algorithm using CRV-Obf'_C instead of \( C \) as input (as outlined above) to compute the challenge secret key \( SK_C \).
5. Run the honest KeyGen algorithm to answer key requests of adversary \( A_2 \).

By the construction of the CRV simulator, the dummy obfuscator CRV-Obf'_C is indistinguishable to adversary \( A_2 \) from the real obfuscator obfuscator CRV-Obf_C except with probability \( \epsilon \).

5.2 Data Hiding.

Achieving relax sim security for the case of data hiding (definition 2.3) is different from the case of function hiding in a subtle but crucial sense. For the case of function hiding, the simulator did not have any restrictions on the queries it could make to the function oracle; that is, given access to oracle \( W_C(\cdot) \), the simulator could query it on any points \( x_i \) of its choice, and in particular on all points \( x_i \in V \). As mentioned above, this is because for the case of function hiding, oracle queries correspond to encryption queries made by an adversary which are unlimited and unrestricted in the public key setting. However, in the case of data privacy, oracle queries by the simulator correspond to secret key queries by the adversary and must be restricted in nature. We will construct an admissible Relax-AD-SIM^{msg} simulator \( S_e \) for the inner product FE scheme of [LOS+10] and show that the simulator is successful as long as the scheme is 1-AD-IND^{msg} secure. Note that 1-AD-IND^{msg} security for inner product FE was shown by Lewko et al. [LOS+10].

Recall that our admissible simulator \( S_e \) is only permitted to make queries in the set \( Q_S \) where \( |Q_S| = \text{poly}(\kappa, 1/\epsilon) \) and \( Q_S \subseteq Q_1 \cup Q'_2 \). We construct \( S_e \) as follows:

1. \( S_e \) runs the honest Setup algorithm and obtains (PK, MSK).
2. \( S_e \) uses the honest KeyGen algorithm to answer the requests of adversary \( A_1 \). \( A_1 \) eventually outputs some \( x \) unknown to \( S_e \). \( S_e \) gets access to function oracle \( U_x(\cdot) \) where \( U_x(C) = C(x) \).
3. \( S_e \) queries the function oracle \( U_x \) for all points \( C_i \in Q_S \) and samples a message \( x' \) such that \( C_i(x') = C_i(x) \forall C_i \in Q_S \). Next, it computes \( CT_{x'} = \text{Encrypt}(x') \) and outputs this as the challenge CT.
4. \( S_e \) runs the honest KeyGen algorithm to answer key requests of adversary \( A_2 \).
5. When \( A_2 \) outputs some bit \( a \), \( S_e \) outputs the same.

\(^6\text{Note that } A_2 \text{ is a predicate adversary not a general adversary.}\)
We show that the above simulator succeeds except with probability $\epsilon$. We will show that if the above simulation fails then we violate $1$-AD-IND$^{\text{msg}}$ of the inner product FE scheme of [LOS+10]. Assume for the sake of contradiction that $\exists (A_1, A_2)$ that cannot be simulated by $S_\epsilon$ above. We will build an IND adversary $(B_1, B_2)$ from $(A_1, A_2)$ and $S_\epsilon$ as described below. Note that a description of the set $Q_S$ is provided along with $S_\epsilon$.

1. Initializing $B_1$: The public key $PK$ is provided to $B_1$ by the Setup algorithm.
2. Constructing $B_1$: $B_1$ runs $A_1$, answering $A_1$'s queries using its own oracle. $A_1$ eventually outputs some $x$. $B_1$ runs $S_\epsilon$, answering its oracle queries $C_i$ by simply computing $C_i(x)$ (since it knows $x$). $S_\epsilon$ chooses $x'$ so that $C_i(x') = C_i(x) \forall C_i \in Q_S$, which is learnt by $B_1$. $B_1$ outputs $(x, x')$ as the IND challenge messages.
3. Constructing $B_2$: Given $CT^* \in \{CT_x, CT_{x'}\}$, $B_2$ runs $A_2(CT^*)$, answering $A_2$'s queries using its own oracle. Note that for all queries $C_i$ of $A_2$, it holds that $C_i(x) = C_i(x')$ hence $B_2$ is an admissible adversary. $A_2$ eventually outputs a bit $\alpha$ which is output by $B_2$.

By assumption, $A_2$ and hence $B_2$ can distinguish between $CT_x$ and $CT_{x'}$ with non-negligible advantage. Thus we built an admissible IND adversary $(B_1, B_2)$ contradicting the $1$-AD-IND$^{\text{msg}}$ security of the inner product FE scheme.

Remark 1. We remark that for the inner product FE scheme of [LOS+10], $1$-AD-IND$^{\text{msg}}$ security and Relax-AD-SIM$^{\text{msg}}$ security imply one another by the result above and Theorem 3.1.

Achieving many-challenge Relax-AD-SIM security. Since our simulator simply runs the honest Setup algorithm to generate the public parameters and the honest KeyGen algorithm to answer key queries, a simple hybrid argument shows that our simulation composes, and we achieve many-challenge key/CT, adaptive relax sim security.

6 Conclusions

In this work, we provided a unified treatment of simulation based data and function privacy for functional encryption. We showed how to relax the requirements on the simulator to bypass known impossibilities in the standard model for inner product FE. We proposed a new simulation based definition Relax-AD-SIM which we showed to be at least as strong as all the existing achievable definitions in FE literature. Our definition is counter-intuitive and seems too weak at face value – thus, the fact that it implies seemingly much stronger, accepted definitions is valuable in furthering our understanding of security definitions of FE.

References


[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In EUROCRYPT, pages 523–552, 2010. 1


[Fre10] David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In EUROCRYPT, pages 44–61, 2010. 9


[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In CRYPTO, 2013. 1, 2


[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In TCC, pages 457–473, 2009. 1, 2, 3, 17, 18

In this section we provide the notation and preliminaries used in the rest of the paper. We begin by defining some standard notation. We say that a function $f : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$ is negligible if $f(\kappa) \in \kappa^{-\omega(1)}$. For two distributions $D_1$ and $D_2$ over some set $\Omega$ we define the statistical distance $SD(D_1, D_2)$ as

$$SD(D_1, D_2) = \frac{1}{2} \sum_{x \in \Omega} |Pr_{D_1}[x] - Pr_{D_2}[x]|$$

We say that two distribution ensembles $D_1(\kappa)$ and $D_2(\kappa)$ are statistically close or statistically indistinguishable if $SD(D_1(\kappa), D_2(\kappa))$ is a negligible function of $\kappa$.

We say that two distribution ensembles $D_1(\kappa), D_2(\kappa)$ are computationally indistinguishable, denoted by $\approx$, if for all probabilistic polynomial time turing machines $A$,

$$|Pr[A(1^\kappa, D_1(\kappa)) = 1] - Pr[A(1^\kappa, D_2(\kappa)) = 1]| = \text{negl}(\kappa)$$

### A.1 Definition: Functional Encryption

Let $\mathcal{X} = \{X_\kappa\}_{\kappa \in \mathbb{N}}$ and $\mathcal{Y} = \{Y_\kappa\}_{\kappa \in \mathbb{N}}$ denote ensembles where each $X_\kappa$ and $Y_\kappa$ is a finite set. Let $\mathcal{C} = \{C_\kappa\}_{\kappa \in \mathbb{N}}$ denote an ensemble where each $C_\kappa$ is a finite collection of circuits, and each circuit $C \in C_\kappa$ takes as input a string $x \in X_\kappa$ and outputs $C(x) \in Y_\kappa$.

A functional encryption scheme $\mathcal{FE}$ for $\mathcal{C}$ consists of four algorithms $\mathcal{FE} = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ defined as follows.

- **Setup($1^\kappa$)** is a p.p.t. algorithm takes as input the unary representation of the security parameter and outputs the master public and secret keys ($PK, MSK$).
- **KeyGen(MSK, $C$)** is a p.p.t. algorithm that takes as input the master secret key $MSK$ and a circuit $C \in C_\kappa$ and outputs a corresponding secret key $SK_C$.
- **Encrypt($PK, x$)** is a p.p.t. algorithm that takes as input the master public key $PK$ and an input message $x \in X_\kappa$ and outputs a ciphertext $CT$.
- **Decrypt($SK_C, CT$)** is a deterministic algorithm that takes as input the secret key $SK_C$ and a ciphertext $CT$ and outputs $C(x)$.

**Definition A.1 (Correctness).** A functional encryption scheme $\mathcal{FE}$ is correct if for all $C \in C_\kappa$ and all $x \in X_\kappa$,

$$\Pr\left[\left(\begin{array}{l}
PK, MSK \leftarrow \text{Setup}(1^\kappa); \\
\text{Decrypt(\text{KeyGen}(MSK, C), Encrypt(PK, x))} \neq C(x)
\end{array}\right)\right] = \text{negl}(\kappa)$$

where the probability is taken over the coins of Setup, KeyGen, and Encrypt.

**Remark 2.** A functional encryption scheme $\mathcal{FE}$ may permit some intentional leakage of information such as the length of the message $|x|$ or function $|C|$, that is leaked in any public key encryption scheme. This is captured by [BSW11] via the “empty” key, by [AGVW13] by giving this information to the simulator directly and by [BF13] by restricting to adversaries who do not trivially break the system by issuing challenges that differ in such leakage. We use the approach of [AGVW13] and pass on any intentionally leaked information directly to the simulator.
A.2 Data Privacy

In this section we recap known definitions of data privacy for FE.

Indistinguishability Based Data Privacy. In this section we define the standard IND based definition for data privacy in FE.

Definition A.2 (NA-INDmsg and AD-INDmsg-Security). Let $FE$ be a functional encryption scheme for a family of circuits $C$. For every p.p.t. adversary $A = (A_1, A_2)$, consider the following two experiments:

$$
\begin{array}{l}
\text{exp}^{(0)}_{FE,A}(1^\kappa): \\
1: (PK, MSK) \leftarrow \text{Setup}(1^\kappa) \\
2: (x_0, x_1, st) \leftarrow A_1^{\text{KeyGen}(MSK, \cdot)}(PK) \\
3: CT \leftarrow \text{Encrypt}(PK, x_0) \\
4: b \leftarrow A_2^{O(MSK, \cdot)}(PK, CT, st) \\
5: \text{Output } b
\end{array}
\begin{array}{l}
\text{exp}^{(1)}_{FE,A}(1^\kappa): \\
1: (PK, MSK) \leftarrow \text{Setup}(1^\kappa) \\
2: (x_0, x_1, st) \leftarrow A_1^{\text{KeyGen}(MSK, \cdot)}(PK) \\
3: CT \leftarrow \text{Encrypt}(PK, x_1) \\
4: b \leftarrow A_2^{O(MSK, \cdot)}(PK, CT, st) \\
5: \text{Output } b
\end{array}$$

Define an admissible adversary $A = (A_1, A_2)$ as one such that for each oracle query $C$ of $A$, $C(x_0) = C(x_1)$. We distinguish between two cases of the above experiment:

1. The adaptive experiment, where the oracle $O(MSK, \cdot) = \text{KeyGen}(MSK, \cdot)$: the functional encryption scheme $FE$ is said to be indistinguishable-secure for one message against adaptive adversaries (1-AD-INDmsg-secure, for short) if for every admissible p.p.t. admissible adversary $A = (A_1, A_2)$, the advantage of $A$ defined as below is negligible in the security parameter $\kappa$:

$$\text{Adv}_{FE,A}(\kappa) = |\Pr[\text{exp}^{(0)}_{FE,A}(1^\kappa) = 1] - \Pr[\text{exp}^{(1)}_{FE,A}(1^\kappa) = 1]|$$

where the probability is over the random coins of the algorithms of the scheme $FE$ and that of $A$.

2. The non-adaptive experiment, where the oracle $O(MSK, \cdot)$ is the “empty oracle” that returns nothing: the functional encryption scheme $FE$ is said to be indistinguishable-secure for one message against non-adaptive adversaries (1-NA-INDmsg-secure, for short) if for every admissible p.p.t. adversary $A = (A_1, A_2)$, the advantage of $A$ defined as above is negligible in the security parameter $\kappa$.

We do not distinguish between one and many message security since this definition composes [GVW12].

Simulation Based Data Privacy. In this section we define the standard notions 1-AD-SIMmsg, 1-NA-SIMmsg) for simulation based security for data privacy.

Definition A.3 (Real and Ideal experiments.). Let $FE$ be a functional encryption scheme for a circuit family $C$. Consider a p.p.t. adversary $A = (A_1, A_2)$ and a stateful p.p.t. simulator $S$.

Let $U_x(\cdot)$ denote a universal oracle, such that $U_x(C) = C(x)$. Consider the following two experiments:

$$
\begin{array}{l}
\text{real msg } \text{exp}^{\text{real msg}}_{FE,A}(1^\kappa): \\
1: (PK, MSK) \leftarrow \text{Setup}(1^\kappa) \\
2: (\vec{x}, st) \leftarrow A_1^{\text{KeyGen}(MSK, \cdot)}(PK) \\
3: CT \leftarrow \text{Encrypt}(PK, \vec{x}) \\
4: \alpha \leftarrow A_2^{O(MSK, \cdot)}(PK, CT, st) \\
5: \text{Output } (\vec{x}, \alpha)
\end{array}
\begin{array}{l}
\text{ideal msg } \text{exp}^{\text{ideal msg}}_{FE,S}(1^\kappa): \\
1: PK \leftarrow S_1(1^\kappa) \\
2: (\vec{x}, st) \leftarrow A_1^{S_1(\cdot)}(PK) \\
3: CT \leftarrow S_2(U_x)(1^\kappa, 1^{|x|}) \\
4: \alpha \leftarrow A_2^{O(\cdot)}(PK, CT, st) \\
5: \text{Output } (\vec{x}, \alpha)
\end{array}$$

$^7$One can replace a stateful simulator by a regular (stateless) simulator that outputs a state $st_s$ upon each invocation which is carried over to its next invocation.
Definition A.4 (Admissible simulator). We call a stateful simulator algorithm $S$ admissible if it makes exactly the same circuit queries to its oracle $U_x(\cdot)$ as the real world adversary $A$ makes to the key derivation oracle, and runs in time poly($\kappa$).

Simulation security can be classified into the following two types:

1. The adaptive case which allows both pre-challenge and post-challenge queries, i.e. where:

   - the oracle $O(MSK, \cdot) = \text{KeyGen}(MSK, \cdot)$ and
   - the oracle $O'(\cdot)$ is the simulator, namely $S^{U_x(\cdot)}(\cdot)$

2. The non-adaptive case, which only allows pre-challenge queries, i.e. where the oracles $O(MSK, \cdot)$ and $O'(\cdot)$ are both the “empty oracles” that return nothing.

Definition A.5 (1-NA-SIM$^{\text{msg.}}$ and 1-AD-SIM$^{\text{msg.}}$ Security). The functional encryption scheme $\mathcal{FE}$ is then said to be simulation-secure for one message against adaptive (resp. non-adaptive) adversaries (1-AD-SIM$^{\text{msg.}}$ (resp. 1-NA-SIM) secure, for short) if there is an admissible stateful p.p.t. simulator $S$ such that for every p.p.t. adversary $A = (A_1, A_2)$, the following two distributions above are computationally indistinguishable.

$$\left\{ \text{exp}_{\mathcal{FE}}^{\text{real msg}}(1^\kappa) \right\}_{\kappa \in \mathbb{N}} \approx \left\{ \text{exp}_{\mathcal{FE}, S}^{\text{ideal msg}}(1^\kappa) \right\}_{\kappa \in \mathbb{N}}$$

Remark 3. Our definition is identical to that provided in [AGVW13] and stronger than that provided in [BSW11], because in [BSW11], the simulator is allowed to “rewind” the adversary. In the above definition however, the simulator is forced to commit to the ciphertext before $A_2$ is invoked, thus forcing the simulator to be “straight-line”. Note that for the adaptive case (where the adversary is allowed to make key queries post-challenge), composition does not hold, i.e. 1-AD-SIM$^{\text{msg.}}$ does not imply many-AD-SIM$^{\text{msg.}}$ [GVW12].

A.3 Function Privacy

Symmetric Key Setting. Function hiding was first considered by Shen et. al. in the symmetric key setting [SSW09]. They formalized indistinguishability based security notions for achieving function (and data) hiding as follows.

**Single challenge security:** Let $\mathcal{FE}_{\text{PrvSC}}$ be a private key functional encryption scheme for a family of circuits $C: \mathcal{X} \rightarrow \mathcal{Y}$. For a p.p.t. adversary $A = (A_1, A_2)$, consider the following two experiments:

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{exp}^{(0)}_{\text{PrvSC}, A}(1^\kappa)$</td>
<td>$\text{SK} \leftarrow \text{Setup}(1^\kappa)$; $(t, Y_0^0, Y_1^0, st) \leftarrow A_1^{\text{KeyGen}(SK, \cdot), \text{Encrypt}(SK, \cdot)}$; If $t=0$, $\text{chal} \leftarrow \text{Encrypt}(SK, Y_0^0)$; If $t=1$, $\text{chal} \leftarrow \text{KeyGen}(SK, Y_0^0)$; $b \leftarrow A_2^{\text{KeyGen}(SK, \cdot), \text{Encrypt}(SK, \cdot)}(\text{chal}, st)$; Output $b$</td>
</tr>
<tr>
<td>$\text{exp}^{(1)}_{\text{PrvSC}, A}(1^\kappa)$</td>
<td>$\text{SK} \leftarrow \text{Setup}(1^\kappa)$; $(t, Y_0^t, Y_1^t, st) \leftarrow A_1^{\text{KeyGen}(SK, \cdot), \text{Encrypt}(SK, \cdot)}$; If $t=0$, $\text{chal} \leftarrow \text{Encrypt}(SK, Y_0^0)$; If $t=1$, $\text{chal} \leftarrow \text{KeyGen}(SK, Y_1^1)$; $b \leftarrow A_2^{\text{KeyGen}(SK, \cdot), \text{Encrypt}(SK, \cdot)}(\text{chal}, st)$; Output $b$</td>
</tr>
</tbody>
</table>

Here $t \in \{0, 1\}$ represents ciphertext challenge when set to zero, and key challenge when set to one. We call an adversary admissible in this setting if for every circuit query $C$ it makes to the oracle, it holds that $C(Y_0^0) = C(Y_1^0)$, and for every ciphertext query $x$, it holds that $Y_0^1(x) = Y_1^1(x)$. 

17
Definition A.6. The private key functional encryption scheme $\mathcal{FE}_{\text{PrvSC}}$ is said to be single challenge IND secure if for every admissible p.p.t. adversary $A = (A_1, A_2)$, the advantage of $A$ defined as below is negligible in the security parameter $\kappa$:  

$$\text{Adv}_{\text{PrvSC}, A}(\kappa) = |\Pr[\exp^{(0)}_{\text{PrvSC}, A}(\kappa) = 1] - \Pr[\exp^{(1)}_{\text{PrvSC}, A}(\kappa) = 1]|,$$

where the probability is over the random coins of the algorithms of $\mathcal{FE}_{\text{PrvSC}}$ and that of $A$.

**Full security:** We now define a stronger notion of security called full security. Let $\mathcal{FE}_{\text{PrvFS}}$ be a private key functional encryption scheme for a family of circuits $\mathcal{C} : \mathcal{X} \rightarrow \mathcal{Y}$. For a p.p.t. adversary $A$, consider the following two experiments:

\[
\begin{array}{c|c}
\text{exp}^{(0)}_{\text{PrvFS}, A}(\kappa) & \text{exp}^{(1)}_{\text{PrvFS}, A}(\kappa) \\
1: \text{SK} \leftarrow \text{Setup}(1^\kappa) & 1: \text{SK} \leftarrow \text{Setup}(1^\kappa) \\
2: b \leftarrow A^{\text{KeyGen}_0(\text{SK}, \cdot), \text{Encrypt}_0(\text{SK}, \cdot)} & 2: b \leftarrow A^{\text{KeyGen}_1(\text{SK}, \cdot), \text{Encrypt}_1(\text{SK}, \cdot)} \\
3: \text{Output } b & 3: \text{Output } b \\
\end{array}
\]

where KeyGen$_0(\text{SK}, C_0, C_1) = \text{KeyGen}(\text{SK}, C_0)$ and Encrypt$_0(\text{SK}, x_0, x_1) = \text{Encrypt}(\text{SK}, x_0)$ for all $C_0, C_1 \in \mathcal{C}$, $x_0, x_1 \in \mathcal{X}$ and $b \in \{0, 1\}$. In the above experiment, an adversary is said to be admissible if for every pair of key query $(C_0, C_1)$ and every pair of ciphertext query $(x_0, x_1)$, it holds that $C_0(x_0) = C_1(x_1)$.

Definition A.7. The private key functional encryption scheme $\mathcal{FE}_{\text{PrvFS}}$ is said to be fully secure if for every admissible p.p.t. adversary $A$, the advantage of $A$ defined as below is negligible in the security parameter $\kappa$:  

$$\text{Adv}_{\text{PrvFS}, A}(\kappa) = |\Pr[\exp^{(0)}_{\text{PrvFS}, A}(\kappa) = 1] - \Pr[\exp^{(1)}_{\text{PrvFS}, A}(\kappa) = 1]|,$$

where the probability is over the random coins of the algorithms of $\mathcal{FE}_{\text{PrvFS}}$ and that of $A$.

It is easy to see that full security implies single challenge security. For the case of inner product predicates, [SSW09] have shown that a single challenge secure scheme can be used to obtain a scheme that is fully secure (but less efficient).

**Public Key Setting.** In the public key setting, function privacy is much trickier to formulate since an adversary possessing a secret key $SK_C$ can encrypt any message $x$ of her choice and run the legitimate decryption procedure to learn $C(x)$.

Recently Boneh et al. put forth a definition for function privacy in the public key setting [BRS13a]. This notion considers adversaries that are given public parameters of the system and can interact with a real or random function privacy oracle. This oracle takes as input any adversarially-chosen distribution $D$ with “sufficient” entropy (formalized below) over vectors of functions, and outputs secret keys either for functions sampled from $D$ or sampled uniformly. Adversaries are allowed to adaptively interact with the real-or-random oracle, for any polynomial number of queries, as long as the distributions have a certain amount of min-entropy. At the end of the interaction, adversaries should have only a negligible probability of distinguishing between the real or random modes of the oracle.

The only requirement on $D$ is that it have “sufficient” entropy, where the notion of sufficient depends on the underlying functionality of the FE scheme. For example, as observed by [BRS13a], for the case of IBE, it is sufficient if identities are picked from a distribution with $\omega(\log \kappa)$. For inner product FE schemes however, the vectors need to come from a block source so that every element has entropy given the previous elements. We will call such distributions feasible entropy distributions.
Definition A.8 (Real or Random function privacy oracle). The real-or-random function-privacy oracle \( \text{ReOrRand}^\text{Fn} \) takes as input triplets of the form \((\text{mode}, \text{MSK}, \mathcal{D})\), where \(\text{mode} \in \{\text{real, rand}\}\), \(\text{MSK}\) is the master secret key, and \(\mathcal{D}\) is a circuit representing a feasible entropy distribution over \(\mathcal{F}\). If \(\text{mode} = \text{real}\) then the oracle samples \(f \leftarrow \mathcal{D}\) and if \(\text{mode} = \text{rand}\) then it samples \(f \leftarrow \mathcal{F}\) uniformly. It then invokes the algorithm \(\text{KeyGen} (\text{MSK}, \cdot)\) on \(f\) for outputting a secret key \(\text{SK}_f\).

Definition A.9 (Function Privacy Adversary). A legitimate function privacy adversary \(A\) is an algorithm that is given as input a pair \((1^\kappa, \text{PK})\) and oracle access to \(\text{ReOrRand}^\text{Fn}(\text{mode}, \text{MSK}, \cdot)\) for some \(\text{mode} \in \{\text{real, rand}\}\) and to \(\text{KeyGen}(\text{MSK}, \cdot)\). It is required that all of \(A\)'s queries to \(\text{ReOrRand}^\text{Fn}\) come from a feasible entropy distribution.

We may view \(A = (A_1, A_2)\) where \(A_1\) makes key queries to the oracle \(\text{ReOrRand}^\text{Fn}\) and outputs a feasible entropy distribution \(\mathcal{D}\) to the oracle and some state \(st\), and \(A_2\) receives \(st\) as input, and may make additional key queries to the oracle before it finally outputs its guess bit \(b\). Note that though \(A\) can make several key queries, it only outputs a single challenge distribution in the above definition.

Definition A.10 (Function Private Encryption). A public key Functional Encryption scheme \(\mathcal{FE}\) is \(\text{ReOrRand}^\text{Fn}\) function private if for any PPT legitimate adversary \(A\), it holds that the advantage of \(A\) defined as below is negligible in the security parameter \(\kappa\):

\[
\text{Adv}_{\mathcal{FE}, A}(\kappa) = \left| \Pr\left[ \exp_{\mathcal{FE}, A}(1^\kappa) = 1 \right] - \Pr\left[ \exp_{\mathcal{FE}, A}^\text{rand fn}(1^\kappa) = 1 \right] \right| \leq \text{negl}
\]

where the probability is over the random coins of the algorithms of the scheme \(\mathcal{FE}\) and that of \(A\) and for each \(\text{mode} \in \{\text{real, rand}\}\) and \(\kappa \in \mathbb{N}\) the experiment \(\exp_{\mathcal{FE}, A}^\text{real fn}(1^\kappa)\) is defined as follows:

1. \((\text{PK}, \text{MSK}) \leftarrow \text{Setup}(1^\kappa)\)
2. \(b \leftarrow A^{\text{ReOrRand}^\text{Fn}(\text{mode}, \text{MSK}, \cdot), \text{KeyGen}(\text{MSK}, \cdot)}(1^\kappa, \text{PK})\)
3. Output \(b\)

If the above holds for a computationally bounded (resp., unbounded) function privacy adversary making a polynomial number of queries to the \(\text{ReOrRand}^\text{Fn}\) oracle, then the scheme is computationally (resp. statistically) \(\text{ReOrRand}^\text{Fn}\) function private.
Correctness of Public Key Inner Product Scheme

For any, \( SK_v, CT_x \) the pairing evaluations in the decryption part of our scheme proceed as follows. Terms that are marked (\( \times \)) are ones that we do not care about:

\[
\tilde{e}(C_0, K) = \tilde{e}\left((sP \cdot \bar{b}_1), \left(-\sum_{i=1}^{n} H_i \cdot \delta_i \cdot \bar{b}_i + Q_6 \cdot \bar{b}_2 + R_5 \cdot \bar{b}_3\right)\right)
\]

\[
= (-sP \sum_{i=1}^{n} H_i \delta_i) \cdot (\bar{b}_1^T \cdot \bar{b}_1) + (\times)(\bar{b}_1^T \cdot \bar{b}_2) + (\times)(\bar{b}_1^T \cdot \bar{b}_3)
\]

\[
= \psi(-sP \sum_{i=1}^{n} H_i \delta_i) \quad \text{(by Equation 1)}
\]

\[
\tilde{e}(C_i, K_i) = \tilde{e}\left(\left(s(H \cdot \bar{b}_1 + R_i \cdot \bar{b}_3) + \alpha \cdot x_i \cdot (Q \cdot \bar{b}_2 + R_0 \cdot \bar{b}_3) + r_i \cdot \bar{b}_3\right), \left(\delta_i \cdot P \cdot \bar{b}_1 + Q \cdot \zeta \cdot v_i \cdot \bar{b}_2\right)\right)
\]

\[
= (sH_i \delta_i P)\bar{b}_1^T \bar{b}_1 + (\alpha x_i Q \cdot Q \zeta v_i)\bar{b}_1^T \bar{b}_2 + (\times)(\bar{b}_1^T \cdot \bar{b}_2 + \bar{b}_1^T \cdot \bar{b}_3 + \bar{b}_3^T \cdot \bar{b}_1 + \bar{b}_3^T \cdot \bar{b}_2)
\]

\[
= \psi(sH_i \delta_i P + \alpha \zeta Q^2 x_i v_i) \quad \text{(by Equation 1)}
\]

Thus,

\[
\tilde{e}(C_0, K) \cdot \prod_{i=1}^{n} \tilde{e}(C_i, K_i)
\]

\[
= \psi(-sP \sum_{i=1}^{n} H_i \delta_i) + \sum_{i=1}^{n} (\psi(sH_i \delta_i P + \alpha \zeta Q^2 x_i v_i))
\]

\[
= \psi Q^2 \alpha \zeta (\sum_{i=1}^{n} x_i v_i)
\]

When \( \sum_{i=1}^{n} x_i v_i \) is 0 mod \( p \), the final answer is always the identity element of the target group and when it is not, the answer evaluates to a random element in the target group (as \( \psi, Q, \alpha, \zeta \leftarrow \mathbb{Z}_p \)).