Multi-instance Security and Its Application to Password-Based Cryptography

Mihir Bellare[1], Thomas Ristenpart[2], and Stefano Tessaro[3]

[1] Department of Computer Science & Engineering, University of California San Diego
cseweb.ucsd.edu/~mihir/
[2] Department of Computer Sciences, University of Wisconsin - Madison
pages.cs.wisc.edu/~rist/
[3] CSAIL, Massachusetts Institute of Technology
people.csail.mit.edu/tessaro/

Abstract. This paper develops a theory of multi-instance (mi) security and applies it to provide the first proof-based support for the classical practice of salting in password-based cryptography. Mi-security comes into play in settings (like password-based cryptography) where it is computationally feasible to compromise a single instance, and provides a second line of defense, aiming to ensure (in the case of passwords, via salting) that the effort to compromise all of some large number m of instances grows linearly with m. The first challenge is definitions, where we suggest LORX-security as a good metric for mi security of encryption and support this claim by showing it implies other natural metrics, illustrating in the process that even lifting simple results from the si setting to the mi one calls for new techniques. Next we provide a composition-based framework to transfer standard single-instance (si) security to mi-security with the aid of a key-derivation function. Analyzing password-based KDFs from the PKCS#5 standard to show that they meet our indifferentiability-style mi-security definition for KDFs, we are able to conclude with the first proof that per password salts amplify mi-security as hoped in practice. We believe that mi-security is of interest in other domains and that this work provides the foundation for its further theoretical development and practical application.

1 Introduction

This paper develops a theory of *multi-instance security* and applies it to support practices in password-based cryptography.

Background. Password-based encryption (PBE) in practice is based on the PKCS#5 (equivalently, RFC 2898) standard [32]. It encrypts a message M under a password pw by picking a random s-bit *salt* sa, deriving a key $L \leftarrow \mathsf{KD}(pw\|sa)$ and returning $C' \leftarrow C\|sa$ where $C \leftarrow\!\!\$\ \mathcal{E}(L, M)$. Here $\mathcal{E}$ is a symmetric encryption scheme, typically an IND-CPA AES mode of operation, and key-derivation function (KDF) KD: $\{0,1\}^* \to \{0,1\}^n$ is the c-fold iteration $\mathsf{KD} = H^c$ of a cryptographic hash function H: $\{0,1\}^* \to \{0,1\}^n$. However, passwords are often poorly chosen [29], falling within a set D called a "dictionary" that is small

R. Safavi-Naini and R. Canetti (Eds.): CRYPTO 2012, LNCS 7417, pp. 312–329, 2012.

enough to exhaust. A brute-force attack now recovers the target password pw (thereby breaking the ind-cpa security of the encryption) using cN hashes where $N = |D|$ is the size of the dictionary.Increasing c increases this effort, explaining the role of this iteration count, but c cannot be made too large without adversely impacting the performance of PBE.

Consider now m users, the i-th with password pw_i. If the salt is absent ($s = 0$), the number of hashes for the brute force attack to recover all m passwords remains around cN, but if s is large enough that salts are usually distinct, it rises to mcN, becoming prohibitive for large m. Salting, thus, aims to make the effort to compromise m target passwords scale linearly in m. (It has no effect on the security of encryption under any one, particular target password.)

New directions. This practice, in our view, opens a new vista in theoretical cryptography, namely to look at the multi-instance (mi) security of a scheme. We would seek metrics of security under which an adversary wins when it breaks all of m instances *but not if it breaks fewer*. This means that the mi security could potentially be much higher than the traditional single-instance (si) security. We would have security amplification.

Why do this? As the above discussion of password-based cryptography shows, there are settings where the computational effort t needed to compromise a single instance is feasible. Rather than give up, we provide a second line of defense. We limit the *scale* of the damage, ensuring (in the case of passwords, via the mechanism of salting) that the computational effort to compromise all of m instances is (around) tm and thus prohibitive for large m. We can't prevent the occasional illness, but we can prevent an epidemic.

We initiate the study of multi-instance security with a foundational treatment in two parts. The first part is agnostic to whether the setting is password-based or not, providing definitions for different kinds of mi-security of encryption and establishing relations between them, concluding with the message that what we call LORX-security is a good choice. The second part of our treatment focuses on password-based cryptography, providing a modular framework that proves mi-security of password-based primitives by viewing them as obtained by the composition of a mi-secure KDF with a si-secure primitive, and yielding in particular the first proof that salting works as expected to increase multi-instance security under a strong and formal metric for the latter.

Multi-instance security turns out to be challenging both definitionally (providing metrics where the adversary wins on breaking all instances but not fewer) and technically (reductions need to preserve tiny advantages and standard hybrid arguments no longer work). It also connects in interesting ways to security amplification via direct products and xor lemmas, eg. [37,16,19,30,13,27,34,28,35]. (We import some of their methods and export some novel viewpoints.) We believe there are many fruitful directions for future work, both theoretical (pursuing the connection with security amplification) and applied (mi security could be valuable in public-key cryptography where steadily improving attacks are making current security parameters look uncomfortably close to the edge for single-instance security). Let us now look at all this in some more detail.

LORX. We consider a setting with m independent target keys $K_1, \ldots, K_m$. (They may, but need not, be passwords.) In order to show that mi-security grows with m we want a metric (definition) where the adversary wins if it breaks all m instances of the encryption but does not win if it breaks strictly fewer. If "breaking" is interpreted as recovery of the key then such a metric is easily given: it is the probability that the adversary recovers all m target keys. We refer to this as the UKU (Universal Key Unrecoverability) metric. But we know very well that key-recovery is a weak metric of encryption security. We want instead a mi analog of ind-cpa. The first thing that might come to mind is multi-user security [3,2]. But in the latter the adversary wins (gets an advantage of one) even if it breaks just one instance so the mu-advantage of an adversary can never be less than its si (ind-cpa) advantage. We, in contrast, cannot "give up" once a single instance is broken. Something radically different is needed.

Our answer is LORX (left-or-right xor indistinguishability). Our game picks m independent challenge bits $b_1, \ldots, b_m$ and gives the adversary an oracle $\mathbf{Enc}(\cdot, \cdot, \cdot)$ which given i, M_0, M_1 returns an encryption of M_{b_i} under K_i. The adversary outputs a bit b' and its advantage is $2\Pr[b' = b_1 \oplus \cdots \oplus b_m] - 1$.[1] Why xor? Its well-known "sensitivity" means that even if the adversary figures out $m - 1$ of the challenge bits, it will have low advantage unless it also figures out the last. This intuitive and historical support is strengthened by the relations, discussed below, that show that LORX implies security under other natural metrics.

Relations. The novelty of multi-instance security prompts us to step back and consider a broad choice of definitions. Besides UKU and LORX, we define RORX (real-or-random xor indistinguishability, a mi-adaptation of the si ROR notion of [4]) and a natural AND metric where the challenge bits $b_1, \ldots, b_m$ and oracle $\mathbf{Enc}(\cdot, \cdot, \cdot)$ are as in the LORX game but the adversary output is a vector $(b'_1, \ldots, b'_m)$ and its advantage is $\Pr[(b'_1, \ldots, b'_m) = (b_1, \ldots, b_m)] - 2^{-m}$. The relations we provide, summarized in Figure 1, show that LORX emerges as the best choice because it implies all the others with tight reductions. Beyond that, they illustrate that the mi terrain differs from the si one in perhaps surprising ways, both in terms of relations and the techniques needed to establish them.

Thus, in the si setting, LOR and ROR are easily shown equivalent up to a factor 2 in the advantages [4]. It continues to be true that LORX easily implies RORX but the hybrid argument used to prove that ROR implies LOR [4] does not easily extend to the mi setting and the proof that RORX implies LORX is not only more involved but incurs a factor 2^m loss.[2] In the si setting, both

[1] This is a simplification of our actual definition, which allows the adversary to adaptively corrupt instances to reveal the underlying keys and challenge bits. This capability means that LORX-security implies threshold security where the adversary wins if it predicts the xor of the challenge bits of some subset of the instances of its choice. See Section 2 for further justification for this feature of the model.

[2] This (exponential) 2^m factor loss is a natural consequence of the factor of 2 loss in the si case, our bound is tight, and the loss in applications is usually small because advantages are already exponentially vanishing in m. Nonetheless it is not always negligible and makes LORX preferable to RORX.

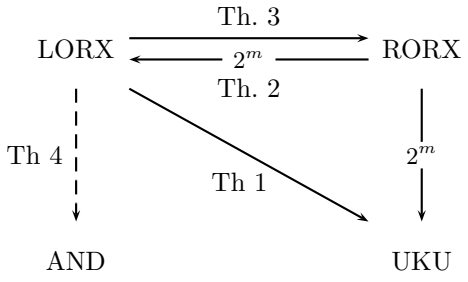

Fig. 1. Notions of multi-instance security for encryption and their relations. LORX (left-or-right xor indistinguishability) emerges as the strongest, tightly implying RORX (real-or-random xor indistinguishability) and UKU (universal key-unrecoverability). The dashed line indicates that under some (mild, usually met) conditions LORX also implies AND. RORX implies LORX and UKU but with a 2^m loss in advantage where m is the number of instances, making LORX a better choice.

LOR and ROR are easily shown to imply KU (key unrecoverability). Showing LORX implies UKU is more involved, needing a boosting argument to ensure preservation of exponentially-vanishing advantages. This reduction is tight but, interestingly, the reduction showing RORX implies UKU is not, incurring a 2^m-factor loss, again indicating that LORX is a better choice. We show that LORX usually implies AND by exploiting a direct product theorem by Unger [35], evidencing the connections with this area. Another natural metric of mi-security is a threshold one, but our incorporation of corruptions means that LORX implies security under this metric.

Mi-security of PBE. Under the LORX metric, we prove that the advantage ϵ' obtained by a time t adversary against m instances of the above PBE scheme $\mathcal{E}'$ is at most $\epsilon + (q/mcN)^m$ (we are dropping negligible terms) where q is the number of adversary queries to RO H and ϵ is the advantage of a time t ind-cpa (si) adversary against $\mathcal{E}$. This is the desired result saying that salting works to provide a second line of defense under a strong mi security metric, amplifying security linearly in the number of instances.

Framework. This result for PBE is established in a modular (rather than ad hoc) way, via a framework that yields corresponding results for any password-based primitive. This means not only ones like password-based message authentication (also covered in PKCS#5) or password-based authenticated encryption (WinZip) but public-key primitives like password-based digital signatures, where the signing key is derived from a password. We view a password-based scheme for a goal as derived by composing a key-derivation function (KDF) with a standard (si) scheme for the same goal. The framework then has the following components. (1) We provide a definition of mi-security for KDFs. (2) We provide composition theorems, showing that composing a mi-secure KDF with a si-secure scheme for a goal results in a mi-secure scheme for that goal. (We will illustrate this for the case of encryption but similar results may be shown for other primitives.) (3) We analyze the iterated hash KDF of PKCS#5 and establish its mi security.

The statements above are qualitative. The quantitative security aspect is crucial. The definition of mi-security of KDFs must permit showing mi-security much higher than si-security. The reductions in the composition theorems must preserve exponentially vanishing mi-advantages. And the analysis of the PKCS#5 KDF must prove that the adversary advantage in q queries to the RO H grows as $(q/cmN)^m$, not merely q/cN. These quantitative constraints represent important technical challenges.

Mi-security of KDFs. We expand on item (1) above. The definition of mi-security we provide for KDFs is a simulation-based one inspired by the indifferentiability framework [26,11]. The attacker must distinguish between the real world and an ideal counterpart. In both, target passwords $pw_1, \ldots, pw_m$ and salts $sa_1, \ldots, sa_m$ are randomly chosen. In the real world, the adversary gets input $(pw_1, sa_1, \mathsf{KD}(pw_1 \| sa_1)), \ldots, (pw_m, sa_m, \mathsf{KD}(pw_m \| sa_1))$ and also gets an oracle for the RO hash function H used by KD. In the ideal world, the input is $(pw_1, sa_1, L_1), \ldots, (pw_m, sa_m, L_m)$ where the keys $L_1, \ldots, L_m$ are randomly chosen, and the oracle is a simulator. The simulator itself has access to a **Test** oracle that will take a guess for a password and tell the simulator whether or not it matches one of the target passwords. Crucially, we require that when the number of queries made by the adversary to the simulator is q, the number of queries made by the simulator to its **Test** oracle is only q/c. This restriction is critical to our proof of security amplification and a source of challenges therein.

Related work. Previous work which aimed at providing proof-based assurances for password-based key-derivation has focused on the single-instance case and the role of iteration as represented by the iteration count c. Our work focuses on the multi-instance case and the roles of both salting and iteration.

The UNIX password hashing algorithm maps a password pw to $E^c_{pw}(0)$ where E is a blockcipher and 0 is a constant. Luby and Rackoff [24] show this is a one-way function when $c = 1$ and pw is a random blockcipher key. (So their result does not really cover passwords.) Wagner and Goldberg [36] treat the more general case of arbitrary c and keys that are passwords, but the goal continues to be to establish one-wayness and no security amplification (meaning increase in security with c) is shown. Boyen [8,9] suggests various ways to enhance security, including letting users pick their own iteration counts.

Yao and Yin [38] give a natural pseudorandomness definition of a KDF in which the attacker gets (K, sa) where K is either $H^c(pw \| sa)$ or a random string of the same length and must determine which. Modeling H as a random oracle (RO) [7] to which the adversary makes q queries, they claim to prove that the adversary's advantage is at most q/cN plus a negligible term. This would establish single-instance security amplification by showing that iteration works as expected to increase attacker effort.[3] However, even though salts are considered,

[3] Unfortunately, we point in [6] to a bug in the proof of [38, Lemma 2.2] and explain why the bound claimed by [38, Theorem 1] is wrong. Beyond this, the proof makes some rather large and not fully justified jumps. The special case $m = 1$ of our treatment will fill these gaps and recover the main claim of [38].

this does not consider multi-instance security let alone establish multi-instance security amplification, and their definition of KDF security does not adapt to allow this. (We use, as indicated above, an indifferentiability-style definition.) In fact the KDF definition of [38] is not even sufficient to establish si security of password-based encryption in the case the latter, as specified in PKCS#5, picks a fresh salt for each message encrypted. Kelsey, Schneier, Hall and Wagner [21] look into the time for password-recovery attacks for different choices of KDFs.

KDFs are for use in non-interactive settings like encryption with WinZip. The issues and questions we consider do not arise with password authenticated key exchange (PAKE) [5,10,14] where definitions already guarantee that the session key may be safely used for encryption. There are no salts and no amplification issues. Abadi and Warinschi [1] provide a si, key-recovery definition for PBE security and connect this with symbolic notions. They do not consider mi security. Dodis, Gennaro, Håstad, Krawczyk and Rabin [12] treat statistically-secure key derivation using hash functions and block ciphers. As discussed in-depth by Kracwzyk [23], these results and techniques aren't useful for password-based KDFs because passwords aren't large enough, let alone have the sufficient amount of min-entropy. Krawczyk [23] also notes that his two-stage KDF approach could be used to build password-based KDFs by replacing the extraction stage with a key-stretching operation. Our general framework may be used to analyze the mi-security of this construction.

Work on direct product theorems and XOR lemmas (eg. [37,15,18,13,27]) has considered the problem of breaking multiple instances of a cryptographic primitive, in general as an intermediate step to amplifying security in the single-instance setting. Mi-Xor-security is used in this way in [13,27].

2 The Multi-instance Terrain

This section defines metrics of mi-secure encryption and explores the relations between them to establish the notions and results summarized in Figure 1. Our treatment intends to show that the mi terrain is different from the si one in fundamental ways, leading to new definitions, challenges and connections.

Syntax. Recall that a symmetric encryption scheme is a triple of algorithms $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. The key generation algorithm $\mathcal{K}$ outputs a key. The encryption algorithm $\mathcal{E}$ takes a key K and a message M and outputs a ciphertext $C \leftarrow\!\!\$\ \mathcal{E}(K, M)$. The deterministic decryption algorithm $\mathcal{D}$ takes K and a ciphertext C to return either a string or $\perp$. Correctness requires that $\mathcal{D}(K, \mathcal{E}(K, M)) = M$ for all M with probability 1 over $K \leftarrow\!\!\$\ \mathcal{K}$ and the coins of $\mathcal{E}$.

To illustrate the issues and choices in defining mi security, we start with key unrecoverability which is simple because it is underlain by a computational game and its mi counterpart is easily and uncontentiously defined. When we move to stronger notions underlain by decisional games, definitions will get more difficult and more contentious as more choices will emerge.

UKU. Single-instance key unrecoverability is formalized via the game $\mathrm{KU}_{\mathsf{SE}}$ where a target key $K \leftarrow\!\!\$\ \mathcal{K}$ is initially sampled, and the adversary $\mathcal{A}$ is given an

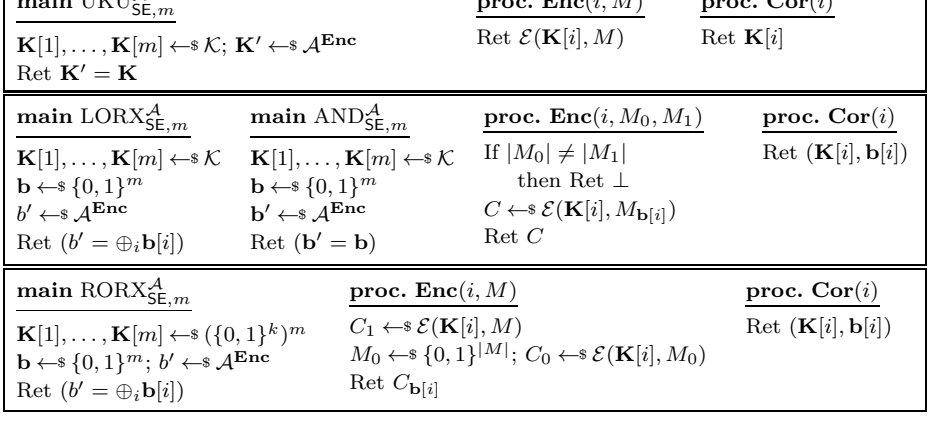

Fig. 2. Multi instance security notions for encryption

oracle **Enc** which, on input M, returns $\mathcal{E}(K, M)$. Finally, the adversary is asked to output a guess K' for the key, and the game returns true if $K = K'$, and false otherwise. An mi version of the game, $\mathrm{UKU}_{\mathsf{SE},m}$, is depicted in Figure 2. It picks an m-vector $\mathbf{K}$ of target keys and the oracle **Enc** now takes i, M to return $\mathcal{E}(\mathbf{K}[i], M)$. The **Cor** oracle gives the adversary the capability of corrupting a user to obtain its target key. The adversary's output guess is also a m-vector $\mathbf{K}'$ and the game returns the boolean $(\mathbf{K} = \mathbf{K}')$, meaning the adversary wins only if it recovers *all* the target keys. (The "U" in "UKU" reflects this, standing for "Universal.") The advantage of adversary $\mathcal{A}$ is $\mathbf{Adv}^{\mathrm{uku}}_{\mathsf{SE},m}(\mathcal{A}) = \Pr[\mathrm{UKU}^{\mathcal{A}}_{\mathsf{SE},m} \Rightarrow \mathsf{true}]$. Naturally, this advantage depends on the adversary's resources. (It could be 1 if the adversary corrupts all instances.) We say that $\mathcal{A}$ is a $(t, \mathbf{q}, q_c)$-adversary if it runs in time t and makes at most $\mathbf{q}[i]$ encryption queries of the form $\mathbf{Enc}(i, \cdot)$ and makes at most q_c corruption queries. Then we let $\mathbf{Adv}^{\mathrm{uku}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{uku}}_{\mathsf{SE},m}(\mathcal{A})$ where the maximum is over all $(t, \mathbf{q}, q_c)$-adversaries.

AND. Single-instance indistinguishabilty for symmetric encryption is usually formalized via left-or-right security [4]. A random bit b and key $K \leftarrow\$ \mathcal{K}$ are chosen, and an adversary $\mathcal{A}$ is given access to an oracle **Enc** that given equal-length messages M_0, M_1 returns $\mathcal{E}(K, M_b)$. The adversary outputs a bit b' and its advantage is $2\Pr[b = b'] - 1$. There are several ways one might consider creating an mi analog. Let us first consider a natural AND-based metric based on game $\mathrm{AND}_{\mathsf{SE},m}$ of Figure 2. It picks at random a vector $\mathbf{b} \leftarrow\$ \{0,1\}^m$ of challenge bits as well as a vector $\mathbf{K}[1],\ldots,\mathbf{K}[m]$ of keys, and the adversary is given access to oracle **Enc** that on input i, M_0, M_1, where $|M_0| = |M_1|$, returns $\mathcal{E}(\mathbf{K}[i], M_{\mathbf{b}[i]})$. Additionally, the corruption oracle **Cor** takes i and returns the pair $(\mathbf{K}[i], \mathbf{b}[i])$. The adversary finally outputs a bit vector $\mathbf{b}'$, and wins if and only if $\mathbf{b} = \mathbf{b}'$. (It is equivalent to test that $\mathbf{b}[i] = \mathbf{b}'[i]$ for all uncorrupted i.) The advantage of adversary $\mathcal{A}$ is $\mathbf{Adv}^{\mathrm{and}}_{\mathsf{SE},m}(\mathcal{A}) = \Pr[\mathrm{AND}^{\mathcal{A}}_{\mathsf{SE},m} \Rightarrow \mathsf{true}] - 2^{-m}$. We say that $\mathcal{A}$

is a $(t, \mathbf{q}, q_c)$-adversary if it runs in time t and makes at most $\mathbf{q}[i]$ encryption queries of the form $\mathbf{Enc}(i, \cdot, \cdot)$ and makes at most q_c corruption queries. Then we let $\mathbf{Adv}^{\mathrm{and}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{and}}_{\mathsf{SE},m}(\mathcal{A})$ where the maximum is over all $(t, \mathbf{q}, q_c)$-adversaries.

This metric has many points in its favor. By (later) showing that security under it is implied by security under our preferred LORX metric, we automatically garner whatever value it offers. But the AND metric also has weaknesses that in our view make it inadequate as the primary choice. Namely, it does not capture the hardness of breaking *all* the uncorrupted instances. For example, an adversary that corrupts instances $1, \ldots, m-1$ to get $\mathbf{b}[1], \ldots, \mathbf{b}[m-1]$, makes a random guess g for $\mathbf{b}[m]$ and returns $(\mathbf{b}[1], \ldots, \mathbf{b}[m-1], g)$ has the high advantage $0.5 - 2^{-m}$ without breaking all instances. We prefer a metric where this adversary's advantage is close to 0.

LORX. To overcome the above issue with the AND advantage, we introduce the XOR advantage measure and use it to define LORX. Game $\mathrm{LORX}_{\mathsf{SE},m}$ of Figure 2 makes its initial choices the same way as game $\mathrm{AND}_{\mathsf{SE},m}$ and provides the adversary with the same oracles. However, rather than a vector, the adversary must output a bit b', and wins if this equals $\mathbf{b}[1] \oplus \cdots \oplus \mathbf{b}[m]$. (It is equivalent to test that $b' = \oplus_{i \in S} \mathbf{b}[i]$ where S is the uncorrupted set.) The advantage of adversary $\mathcal{A}$ is $\mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(\mathcal{A}) = 2\Pr[\mathrm{LORX}^{\mathcal{A}}_{\mathsf{SE},m} \Rightarrow \mathsf{true}] - 1$. We say that $\mathcal{A}$ is a $(t, \mathbf{q}, q_c)$-adversary if it runs in time t and makes at most $\mathbf{q}[i]$ encryption queries of the form $\mathbf{Enc}(i, \cdot, \cdot)$ and makes at most q_c corruption queries. Then we let $\mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(\mathcal{A})$ where the maximum is over all $(t, \mathbf{q}, q_c)$-adversaries. In the example we gave for AND, if an adversary corrupts the first $m-1$ instances to get back $\mathbf{b}[1], \ldots, \mathbf{b}[m-1]$, makes a random guess g for $\mathbf{b}[m]$ and outputs $b' = \mathbf{b}[1] \oplus \cdots \oplus \mathbf{b}[m-1] \oplus g$, it will have advantage 0.

RORX. A variant of the si LOR notion, ROR, was given in [4]. Here the adversary must distinguish between an encryption of a message M it provides and the encryption of a random message of length $|M|$. This was shown equivalent to LOR up to a factor 2 in the advantages [4]. This leads us to define the mi analog RORX and ask how it relates to LORX. Game $\mathrm{RORX}_{\mathsf{SE},m}$ of Figure 2 makes its initial choices the same way as game $\mathrm{LORX}_{\mathsf{SE},m}$. The adversary is given access to oracle $\mathbf{Enc}$ that on input i, M, returns $\mathcal{E}(\mathbf{K}[i], M)$ if $\mathbf{b}[i] = 1$ and otherwise returns $\mathcal{E}(\mathbf{K}[i], M_1)$ where $M_1 \leftarrow\!\!\$\ \{0,1\}^{|M|}$. It also gets the usual $\mathbf{Cor}$ oracle. It outputs a bit b' and wins if this equals $\mathbf{b}[1] \oplus \cdots \oplus \mathbf{b}[m]$. The advantage of adversary $\mathcal{A}$ is $\mathbf{Adv}^{\mathrm{rorx}}_{\mathsf{SE},m}(\mathcal{A}) = 2\Pr[\mathrm{RORX}^{\mathcal{A}}_{\mathsf{SE},m} \Rightarrow \mathsf{true}] - 1$. We say that $\mathcal{A}$ is a $(t, \mathbf{q}, q_c)$-adversary if it runs in time t and makes at most $\mathbf{q}[i]$ encryption queries of the form $\mathbf{Enc}(i, \cdot)$ and makes at most q_c corruption queries. Then we let $\mathbf{Adv}^{\mathrm{rorx}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{rorx}}_{\mathsf{SE},m}(\mathcal{A})$ where the maximum is over all $(t, \mathbf{q}, q_c)$-adversaries.

Discussion. The multi-user security goal from [3] gives rise to a version of the above games without corruptions and where all instances share the same challenge bit b, which the adversary tries to guess. But this does *not* measure mi security, since recovering a single key suffices to learn b.

The above approach extends naturally to providing a mi counterpart to any security definition based on a decisional game, where the adversary needs to guess a bit b. For example we may similarly create mi metrics of CCA security.

Why does the model include corruptions? The following example may help illustrate. Suppose SE is entirely insecure when the key has first bit 0 and highly secure otherwise. (From the si perspective, it is insecure.) In the LORX game, an adversary will be able to figure out around half the challenge bits. If we disallow corruptions, it would still have very low advantage. From the application point of view, this seems to send the wrong message. We want LORX-security to mean that the probability of "large scale" damage is low. But breaking half the instances is pretty large scale. Allowing corruptions removes this defect because the adversary could corrupt the instances it could not break and then, having corrupted only around half the instances, get a very high advantage, breaking LORX-security. In this way, we may conceptually keep the focus on an adversary goal of breaking *all* instances, yet cover the case of breaking some threshold number via the corruption capability.

An alternative way to address the above issue without corruptions is to define threshold metrics where the adversary wins by outputting a dynamically chosen set S and predicting the xor of the challenge bits for the indexes in S. This, again, has much going for it as a metric. But LORX with corruptions, as we define it, will imply security under this metric.

LORX IMPLIES UKU. In the si setting, it is easy to see that LOR security implies KU security. The LOR adversary simply runs the KU adversary. When the latter makes oracle query M, the LOR adversary queries its own oracle with M, M and returns the outcome to the KU adversary. When the latter returns a key K', the LOR adversary submits a last oracle query consisting of a pair M_0, M_1 of random messages to get back a challenge ciphertext C, returning 1 if $\mathcal{D}(K', C) = M_1$ and 0 otherwise. A similar but slightly more involved proof shows that ROR implies KU.

It is important to establish analogs of these basic results in the mi setting, for they function as "tests" for the validity of our mi notions. The following shows that LORX security implies UKU. Interestingly, it is not as simple to establish in the mi case as in the si case. Also, as we will see later, the proof that RORX implies UKU is not only even more involved but incurs a factor 2^m loss, making LORX a better choice as the metric to target in designs.

Theorem 1. [LORX $\Rightarrow$ UKU] *Let $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme with message space $\mathcal{M}$, and let ℓ be such that $\{0,1\}^\ell \subseteq \mathcal{M}$. Then, for all t, q_c, and $\mathbf{q}$, and for all $k \geq 1$,*

$$\mathbf{Adv}^{\mathrm{uku}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) \leq \mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(t', \mathbf{q}', q_c) + m \cdot \left(\frac{1}{2^\ell - 1}\right)^k ,$$

where $t' = t + O(m \cdot k)$, and $\mathbf{q}'[i] = \mathbf{q}[i] + k$ for all $i = 1, \ldots, m$. ∎

The proof is given in [6]. Here, let us stress Theorem 1 surfaces yet another subtlety of the mi setting. At first, it would seem that proving the case $k = 1$

of the theorem is sufficient (this is what usually done in the si case). However, it is crucial to remark that $\mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(t', \mathbf{q}', q_c)$ may be *very* small. For example, it is not unreasonable to expect $2^{-128 \cdot m}$ if SE is secure in the single-instance setting. Yet, assume that $\mathcal{E}$ encrypts 128-bit messages, then we are only able to set $\ell = 128$, in turn making $m/(2^\ell - 1) \approx m \cdot 2^{-128}$ by far the leading term on the right-hand side. The parameter k hence opens the door to fine tuning of the additive extra term at the cost of an additive complexity loss in the reduction. Also note that the reduction in the proof of Theorem 1 is not immediate, as an adversary guessing all the keys in the UKU game with probability ϵ only yields an adversary recovering all the bits $\mathbf{b}[1], \ldots, \mathbf{b}[m]$ in the LORX game with probability ϵ. Just outputting the xor of these bits is not sufficient, as we have to boost the success probability to $\frac{1+\epsilon}{2}$ in order to obtain the desired relation between the two advantage measures.

In analogy to the si setting, UKU does not imply LORX. Just take a scheme $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ encrypting n-bit messages which is UKU-secure, and modify it into a scheme $\mathsf{SE}' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ where $\mathcal{K} = \mathcal{K}'$ and $\mathcal{E}'(K, M) = \mathcal{E}'(K, M) \,\|\, M[0]$, with $M[0]$ being the first bit of M. Clearly, SE' is still UKU-secure but not LORX-secure

As indicated above, a proof that RORX implies UKU is much more involved and incurs a factor 2^m loss. Roughly speaking, this is because in the si case, in the reduction needed to prove that ROR implies KU, the ROR adversary can only simulate the execution of the KU adversary correctly in the case where the bit is 1, i.e., the encryption oracle returns the actual encryption of a message. This results in a factor two loss in terms of advantage. Upon translating this technique to the mi case, the factor 2 becomes 2^m, as *all* bits need to be 1 for the UKU adversary to output the right keys with some guaranteed probability. However, we will not follow this route for the proof of this result. Instead, we can obtain the same result by combining Theorem 2 and Theorem 1.

LORX VERSUS RORX. In the si setting, LOR and ROR are the same up to a factor 2 in the advantage [4]. The LOR implies ROR implication is trivial and ROR implies LOR is a simple hybrid argument. We now discuss the relation between the mi counterparts, namely RORX and LORX, which is both more complex and more challenging to establish.

Theorem 2. [RORX $\Rightarrow$ LORX] *Let* $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a symmetric encryption scheme. For all* m, $t, q_c > 0$, *and all vectors* $\mathbf{q}$ *we have* $\mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) \leq 2^m \cdot \mathbf{Adv}^{\mathrm{rorx}}_{\mathsf{SE},m}(t', \mathbf{q}, q_c)$, *where* $t' = t + \mathcal{O}(1)$. ∎

As discussed in Section 1, the multiplicative factor 2^m is often of no harm because advantages are already exponentially small in m. The factor is natural, being the mi analogue of the factor 2 appearing in the traditional si proof, and examples can be given showing that the bound is tight. The proof of the above is in [6]. The difficulty is adapting the hybrid argument technique to the mi setting. We omit the much simpler proof of the converse:

Theorem 3. [LORX ⇒ RORX] *Let* $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a symmetric encryption scheme. For all* m, $t, q_c > 0$, *and all vectors* $\mathbf{q}$ *we have* $\mathbf{Adv}^{\mathrm{rorx}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) \leq \mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(t', \mathbf{q}, q_c)$, *where* $t' = t + \mathcal{O}(1)$. ∎

LORX implies AND. Intuitively, one might expect AND security to be a *stronger* requirement than LORX security, as the former seems easier to break than the latter. However we show that under a fairly minimal requirement, LORX implies AND. This brings another argument in support of LORX: Even if an application requires AND security, it turns out that proving LORX security is generally sufficient. The following theorem is to be interpreted as follows: In general, if we only know that $\mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c)$ is small, we do not know how to prove $\mathbf{Adv}^{\mathrm{and}}_{\mathsf{SE},m}(t', \mathbf{q}, q_c)$ is also small (for $t' \approx t$), or whether this is true at all. As we sketched above, the reason is that we do not know how to use an adversary $\mathcal{A}$ for which the $\mathrm{AND}_{\mathsf{SE},m}$ advantage is large to construct an adversary for which the $\mathrm{LORX}_{\mathsf{SE},m}$ advantage is large. Still, one would expect that such an adversary *might* more easily yield one for which the $\mathrm{LORX}_{\mathsf{SE},k}$ advantage is sufficiently large, for *some* $k \leq m$. The following theorem uses a probabilistic lemma due to Unger [35] to confirm this intuition.

Theorem 4. *Let* $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a symmetric encryption scheme. Further, let* m, t, $\mathbf{q}$, *and* q_c *be given, and assume that there exist* C, ϵ, *and* γ *such that for all* $1 \leq i \leq m$,

$$\max_{S \subseteq \{1,\ldots,m\}, |S|=i} \mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},i}(t^*_S, \mathbf{q}[S], q_c) \leq C \cdot \epsilon^i + \gamma \ ,$$

where $\mathbf{q}[S]$ *is the projection of* $\mathbf{q}$ *on the components in* S, *and* $t^*_S = t + \mathcal{O}(t_{\mathcal{E}} \cdot \sum_{i \notin S} \mathbf{q}[i])$, *with* $t_{\mathcal{E}}$ *denoting the running time needed for one encryption with* $\mathcal{E}$. *Then,* $\mathbf{Adv}^{\mathrm{and}}_{\mathsf{SE},m}(t, \mathbf{q}, q_c) \leq \gamma + C \cdot \prod_{i=1}^{m}(1 + \epsilon_i)/2$. ∎

We are not able to prove that the converse (AND implies LORX) is true in general, but in the absence of corruptions one can upper bound $\mathbf{Adv}^{\mathrm{lorx}}_{\mathsf{SE},m}(t, \mathbf{q}, 0)$ in terms of $\mathbf{Adv}^{\mathrm{and}}_{\mathsf{SE},m'}(t', \mathbf{q}', 0)$ for $m' \approx 2m$ and t' and $\mathbf{q}'$ being much larger than $t, \mathbf{q}$. The proof, which we omit, follows the lines of the proof of the XOR Lemma from the Direct Product Theorem given by [18], and relies on the Goldreich-Levin theorem [17]. As the loss in concrete security in this reduction is very large, and it only holds in the corruption-free case, we find this an additional argument to support the usage of the LORX metric.

3 Password-Based Encryption via KDFs

We now turn to our main motivating application, that of password based encryption (PBE) as specified in PKCS#5 [32]. The schemes specified there combine a conventional mode of operation (e.g., CBC mode) with a password-based key derivation function (KDF). We start with formalizing the latter.

Password-based KDFs. Formally, a (k, s, c)-KDF is a deterministic map $\mathsf{KD} : \{0,1\}^* \times \{0,1\}^s \to \{0,1\}^k$ that may make use of an underlying ideal primitive.

Here c is the iteration count, which specifies the multiplicative increase in work that should slow down brute force attacks.

PKCS#5 describes two KDFs [32]. We treat the first in detail and discuss the second in [6]. Let $\mathsf{KD1}^H(pw, sa) = H^c(pw \,\|\, sa)$ where H^c is the function that composes H with itself c times. To generalize beyond concatenation, we can define a function $\mathsf{Encode}(pw, sa)$ that describes how to encode its inputs onto $\{0,1\}^*$ with efficiently computable inverse $\mathsf{Decode}(W)$.

PBE schemes. A PBE scheme is just a symmetric encryption scheme where we view the keys as passwords and key generation as a password sampling algorithm. To highlight when we are thinking of key generation as password sampling we will use $\mathcal{P}$ to denote key generation (instead of $\mathcal{K}$). We will also write pw for a key that we think of as a password. Let KD be a (k, s, c)-KDF and let $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme with $\mathcal{K}$ outputting uniformly selected k-bit keys. Then we define the PBE scheme $\mathcal{SE}[\mathsf{KD}, \mathsf{SE}] = (\mathcal{P}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ as follows. Encryption $\overline{\mathcal{E}}(pw, M)$ is done via $sa \xleftarrow{\$} \{0,1\}^s$; $K \leftarrow \mathsf{KD}(pw, sa)$; $C \xleftarrow{\$} \mathcal{E}(K, M)$, returning (sa, C) as the ciphertext. Decryption recomputes the key K by reapplying the KDF and then applies $\mathcal{D}$. If the KDF is $\mathsf{KD1}$ and the encryption scheme is CBC mode, then one obtains the first PBE scheme from PKCS#5 [32].

Password guessing. We aim to show that security of the above constructions holds up to the amount of work required to brute-force the passwords output by $\mathcal{P}$. This begs the question of how we measure the strength of a password sampler. We will formalize the hardness of guessing passwords output by some sampler $\mathcal{P}$ via an adaptive guessing game: It challenges an adversary with guessing passwords adaptively in a setting where the attacker may, also, adaptively learn some passwords via a corruption oracle. Concretely, let $\mathrm{GUESS}_{\mathcal{P},m}$ be the game defined in Figure 3. A (q_t, q_c)-*guessing adversary* is one that makes at most q_t queries to **Test** and q_c queries to **Cor**. An adversary $\mathcal{B}$'s guessing advantage is $\mathbf{Adv}^{\mathrm{guess}}_{\mathcal{P},m}(\mathcal{B}) = \Pr\left[\mathrm{GUESS}^{\mathcal{B}}_{\mathcal{P},m} \Rightarrow \mathsf{true}\right]$. We assume without loss of generality that $\mathcal{A}$ does not make any *pointless queries*: (1) repeated queries to **Cor** on the same value; (2) a query $\mathbf{Test}(i, \cdot)$ following a query of $\mathbf{Cor}(i)$; and (3) a query $\mathbf{Cor}(i)$ after a query $\mathbf{Test}(i, pw)$ that returned true. We also define a variant of the above guessing game that includes salts and allows an attacker to test password-salt pairs against all m instances simultaneously. This will be useful as an intermediate step when reducing to guessing advantage. The game $\mathrm{saGUESS}_{\mathcal{P},m,\rho}$ is shown in Figure 3 and we define advantage via $\mathbf{Adv}^{\text{sa-guess}}_{\mathcal{P},m}(\mathcal{B}) = \Pr\left[\mathrm{saGUESS}^{\mathcal{B}}_{\mathcal{P},m} \Rightarrow \mathsf{true}\right]$. An easy argument proves the following lemma.

Lemma 5. *Let $m, \rho > 0$, let $\mathcal{P}$ be a password sampler and let $\mathcal{A}$ be an (q_t, q_c)-guessing* $\mathrm{GUESS}_{\mathcal{P},m}$ *adversary. Then there is a (q_t, q_c)-guessing* $\mathrm{saGUESS}_{\mathcal{P},m,\rho}$ *adversary $\mathcal{B}$ such that* $\mathbf{Adv}^{\text{sa-guess}}_{\mathcal{P},m,\rho}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{guess}}_{\mathcal{P},m}(\mathcal{B}) + m^2\rho^2/2^s$. □

Samplers with high min-entropy. Even though the guessing advantage precisely quantifies strength of password samplers, good upper bounds in terms of the adversary's complexity and of some simpler relevant parameters of a

main $\mathrm{GUESS}_{\mathcal{P},m}$	**proc. Test**(i, pw)	**proc. Cor**(i)
$\mathbf{pw}[1], \ldots, \mathbf{pw}[m] \leftarrow\!\!\$\ \mathcal{P}$ $\mathbf{pw}' \leftarrow\!\!\$\ \mathcal{B}^{\mathbf{Test},\mathbf{Cor}}$ Ret $\bigwedge_{i=1}^{m}(\mathbf{pw}'[i] = \mathbf{pw}[i])$	If $(pw = \mathbf{pw}[i])$ then Ret **true** Ret $\perp$	Ret $\mathbf{pw}[i]$

main $\mathrm{saGUESS}_{\mathcal{P},m,\rho}$	**proc. Test**(pw, sa)	**proc. Cor**(i)

$\mathbf{pw}[1], \ldots, \mathbf{pw}[m] \leftarrow\!\!\$\ \mathcal{P}$
For $i = 1$ to m do
  For $j = 1$ to ρ do
  $\mathbf{sa}[i,j] \leftarrow\!\!\$\ \{0,1\}^s$
$\mathbf{pw}' \leftarrow\!\!\$\ \mathcal{B}^{\mathbf{Test},\mathbf{Cor}}(\mathbf{sa})$
Ret $\bigwedge_{i=1}^{m}(\mathbf{pw}'[i] = \mathbf{pw}[i])$
	For $i = 1$ to m do
	  For $j = 1$ to ρ do
	  If $(pw, sa) = (\mathbf{pw}[i], \mathbf{sa}[i,j])$ then
	  Ret (i,j)
	Ret $(\perp, \perp)$
		Ret $\mathbf{pw}[i]$

Fig. 3. An adaptive password-guessing game

password sampler are desirable. One interesting case is samplers with high min-entropy. Formally, we say that $\mathcal{P}$ has min-entropy μ if for all pw' it holds that $\Pr[pw = pw'] \leq 2^{-\mu}$ over the coins used in choosing $pw \leftarrow\!\!\$\ \mathcal{P}$.

Theorem 6. *Fix $m \geq q_c \geq 0$ and a password sampler $\mathcal{P}$ with min-entropy μ. Let $\mathcal{B}$ be a (q_t, q_c)-adversary for $\mathrm{GUESS}_{\mathcal{P},m}$ making q_i queries of the form $\mathbf{Test}(i,\cdot)$ with $q_t = q_1 + \cdots + q_m$. Let $\delta = q_t/(m2^{\mu})$ and let $\gamma = (m - q_c)/m$. Then $\mathbf{Adv}^{\mathrm{guess}}_{\mathcal{P},m}(\mathcal{B}) \leq e^{-m\Delta(\gamma,\delta)}$ where $\Delta(\gamma,\delta) = \gamma\ln(\frac{\gamma}{\delta}) + (1-\gamma)\ln(\frac{1-\gamma}{1-\delta})$.* □

Using $\Delta(\gamma,\delta) \geq 2(\gamma-\delta)^2$, we see that to win the guessing game for q_c corruptions, $q_t \approx (m-q_c)\cdot 2^{\mu}$ **Test** queries are necessary, and the brute-force attack is optimal. Note that the above bound is the best we expect to prove: Indeed, assume for a moment that we restrict ourselves to adversaries that want to recover a subset of $m - q_c$ passwords, without corruptions, and make q_t/m queries $\mathbf{Test}(i,\cdot)$, for each i, which are independent from queries $\mathbf{Test}(j,\cdot)$ for other $j \neq i$. Then, each individual password is found, independently, with probability at most $q_t/(m\cdot 2^{\mu})$, and if one applies the Chernoff bound, the probability that a subset of size $m-q_c$ of the passwords are retrieved is upper bounded by $e^{-m\Delta(\gamma,\delta)}$. In our case, we have additional challenges: Foremost, queries for each i are not independent. Also, the number of queries may not be the same for each index i. And finally, we allow for corruption queries.

The full proof of Theorem 6 is given in [6]. At a high level, it begins by showing how to move to a simpler setting in which the adversary wins by recovering a subset of the passwords without the aid of a corrupt oracle. The resulting setting is an example of a threshold direct product game. This allows us to apply a *generalized* Chernoff bound due to Panconesi and Srinivasan [31] (see also [20]) that reduces threshold direct product games to (non-threshold) direct product games. Finally, we apply an amplification lemma due to Maurer, Pietrzak, and Renner [25] that yields a direct product theorem for the password guessing game. Let us also note that using the same technique, the better bound $\mathbf{Adv}^{\mathrm{guess}}_{\mathcal{P},m}(\mathcal{B}) \leq (q_t/m2^{\mu})^m$ can be proven for the special case of $(q_t, 0)$-adversaries.

Correlated passwords. By taking independent samples from $\mathcal{P}$ we have captured only the setting of independent passwords. In practice, of course, passwords may be correlated across users or, at least, user accounts. Our results extend to the setting of jointly selecting a vector of m passwords, except of course the analysis of the guessing advantage (whose proof fundamentally relies upon independence). This last only limits our ability to measure, in terms of simpler metrics like min-entropy, the difficulty of a guessing game against correlated passwords. This does not decrease the security proven, as the simulation-based paradigm we introduce below allows one to reduce to the full difficulty of the guessing game.

Simulation-based Security for KDFs.We define an ideal-functionality style notion of security for KDFs. Figure 4 depicts two games. A message sampler $\mathcal{M}$ is an algorithm that takes input a number r and outputs a pair of vectors $(\mathbf{pw}, \mathbf{sa})$ each having r elements and with $|\mathbf{sa}[i]| = s$ for $1 \le i \le r$. A simulator S is a randomized, stateful procedure. It expects oracle access to a procedure **Test** to which it can query a message. Game $\mathrm{Real}_{\mathsf{KD},\mathcal{M},r}$ gives a distinguisher $\mathcal{D}$ the messages and associated derived keys. Also, $\mathcal{D}$ can adaptively query the ideal primitive H underlying KD. Game $\mathrm{Ideal}_{S,\mathcal{M},r}$ gives $\mathcal{D}$ the messages and keys chosen uniformly at random. Now $\mathcal{D}$ can adaptively query a primitive oracle implemented by a simulator S that, itself, has access to a **Test** oracle. Then we define KDF advantage by

$$\mathbf{Adv}^{\mathrm{kdf}}_{\mathsf{KD},\mathcal{M},r}(\mathcal{D}, S) = \Pr\left[\,\mathrm{Real}^{\mathcal{D}}_{\mathsf{KD},\mathcal{M},r} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{Ideal}^{\mathcal{D}}_{S,\mathcal{M},r} \Rightarrow 1\,\right].$$

To be useful, we will require proving that there exists a simulator S such that for any $\mathcal{D}, \mathcal{M}$ pair the KDF advantage is "small".

This notion is equivalent to applying the indifferentiability framework [26] to a particular ideal KDF functionality. That functionality chooses messages according to an algorithm $\mathcal{M}$ and outputs on its honest interface the messages and uniform keys associated to them. On the adversarial interface is the test routine which allows the simulator to learn keys associated to messages. This raises the question of why not just use indifferentiability from a RO as our target security notion. The reasons are two-fold. First, it is not clear that H^c is indifferentiable from a random oracle. Second, even if it were, a proof would seem to require a simulator that makes at least the same number of queries to the RO as it receives from the distinguisher. This rules out showing security amplification due to the iteration count c. Our approach solves both issues, since we will show KDF security for simulators that make one call to **Test** for every c made to it. For example, our simulator for $\mathsf{KD1}$ will only query **Test** if a chain of c hashes leads to the being-queried point X and this chain is not a continuation of some longer chain. We formally capture this property of simulators next.

c-amplifying simulators. Let $\tau = (X_1, Y_1), \ldots, (X_q, Y_q)$ be a (possibly partial) transcript of **Prim** queries and responses. We restrict attention to (k, s, c)-KDFs for which we can define a predicate $\mathsf{final}_{\mathsf{KD}}(X_i, \tau)$ which evaluates to true if there exists exactly one sequence of c indices $j_1 < \cdots < j_c$ such that (1) $j_c = i$, (2) there exist unique (pw, sa) such that evaluating $\mathsf{KD}^H(pw, sa)$ when H is such

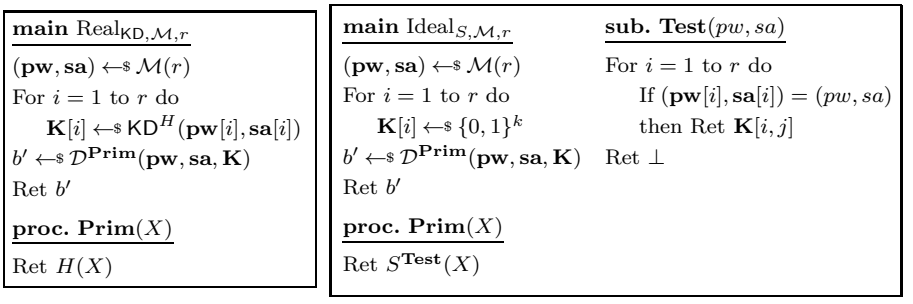

Fig. 4. Games for the simulation-based security notion for KDFs

that $Y_j = H(X_j)$ for $1 \le j \le i$ results exactly in the queries $X_{j_1}, \ldots, X_{j_c}$ in any order where X_i is the last query, and (3) $\mathsf{final}_{\mathsf{KD}}(X_{j_r}, \tau) = \mathsf{false}$ for all $r < c$.

Our simulators only query **Test** on queries X_i for which $\mathsf{final}_{\mathsf{KD}}(X_i, \tau) = \mathsf{true}$; we call such queries KD-*completion queries* and simulators satisfying this are called *c-amplifying*. Note that (3) implies that there are at most q/c total KD-completion queries in a q-query transcript.

Hash-dependent passwords. We do not allow $\mathcal{M}$ access to the random oracle H. This removes from consideration hash-dependent passwords. Our results should extend to cover hash-dependent passwords if one has explicit domain separation between use of H during password selection and during key derivation. Otherwise, an indifferentiability-style approach as we use here will not work due to limitations pointed out in [33]. A full analysis of the hash-dependent password setting would therefore appear to require direct analysis of PBE schemes without taking advantage of the modularity provided by simulation-based approaches.

Security of KD1. For a message sampler $\mathcal{M}$, let $\gamma(\mathcal{M}, r) := \Pr[\exists i \ne j : (\mathbf{pw}[i],\mathbf{sa}[i]) = (\mathbf{pw}[j],\mathbf{sa}[j])]$ where $(\mathbf{pw},\mathbf{sa}) \leftarrow\$ \mathcal{M}(r)$. We prove the following theorem in [6].

Theorem 7. *Fix $r > 0$. Let* KD1 *be as above. There exists a simulator* S *such that for all adversaries $\mathcal{D}$ making q RO queries, of which q_c are chain completion queries, and all message samplers $\mathcal{M}$,*

$$\mathbf{Adv}^{\mathrm{kdf}}_{\mathsf{KD1},\mathcal{M},r}(\mathcal{D}, S) \le 4\,\gamma(\mathcal{P}, r) + \frac{2r^2 + 7\,(2q + rc)^2}{2^n}\ .$$

The simulator S makes at most q_c **Test** *queries, and answers each query in time $O(c)$.* □

Security of PBE. We are now in a position to analyze the security of password based encryption as used in PKCS#5. The following theorem, proved in [6], uses the multi-user left-or-right security notion from [3] whose formalization is recalled in [6]:

Theorem 8. *Let $m \ge 1$, let $\mathcal{SE}[\mathsf{KD},\mathsf{SE}] = (\mathcal{P}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ be the encryption scheme built from an (k, s, c)-KDF* KD *and an encryption scheme $\mathsf{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with*

k-bit keys. Let $\mathcal{A}$ be an adversary making ρ queries to $\boldsymbol{Enc}(i,\cdot,\cdot)$ for each $i \in \{1,\ldots,m\}$ and making at most $q_c < m$ corruption queries. Let S be a c-amplifying simulator. Then there exists message sampler $\mathcal{M}$ and adversaries $\mathcal{D}$, $\mathcal{C}$, and $\mathcal{B}$ such that

$$\mathbf{Adv}^{\mathrm{lorx}}_{\mathcal{SE},m}(\mathcal{A}) \le m \cdot \mathbf{Adv}^{\text{mu-lor}}_{\mathsf{SE},\rho}(\mathcal{C}) + 2 \cdot \mathbf{Adv}^{\mathrm{guess}}_{\mathcal{P},m,\rho}(\mathcal{B}) + 2 \cdot \mathbf{Adv}^{\mathrm{kdf}}_{\mathsf{KD},\mathcal{M},m\rho}(\mathcal{D},S)$$

If $\mathcal{A}$ makes q queries to H, then: $\mathcal{D}$ makes at most q queries to its H oracle; $\mathcal{B}$ makes at most $\lceil q/c \rceil$ queries to **Test** *and at most q_c corruption queries; and $\mathcal{C}$ makes a single query $\boldsymbol{Enc}(i,\cdot,\cdot)$ for each $1 \le i \le \rho$. Moreover, $\mathcal{C}$'s running time equals $t_{\mathcal{A}} + q \cdot t_S$ plus a small, absolute constant, and where $t_{\mathcal{A}}$ is the running time of $\mathcal{A}$, and t_S is the time needed by S to answer a query. Finally, $\gamma(\mathcal{M}, m\rho) \le m^2\rho^2/2^s$.* □

Note that the theorem holds even when SE is only one-time secure (meaning it can be deterministic), which implies that the analysis covers tools such as WinZip (c.f., [22]). In terms of the bound we achieve, Theorem 7 for $\mathsf{KD1}$ shows that an adversary that makes $\mathbf{Adv}^{\mathrm{kdf}}_{\mathsf{KD},\mathcal{P}^*,m\rho}(\mathcal{D},S)$ large requires $q \approx 2^{n/2}$ queries to H, provided salts are large. If H is SHA-256 then this is about 2^{128} work. Likewise, a good choice of SE will ensure that $\mathbf{Adv}^{\text{mu-lor}}_{\mathsf{SE},\mathcal{K},\rho}(\mathcal{C})$ will be very small. Thus the dominating term ends up the guessing advantage of $\mathcal{B}$ against $\mathcal{P}$, which measures its ability to guess $m - q_c$ passwords.

Acknowledgments. Bellare was supported in part by NSF grants CCF-0915675, CNS-0904380 and CNS-1116800. Ristenpart was supported in part by NSF grant CNS-1065134. Tessaro was supported in part by NSF grants CCF-0915675, CCF-1018064, and DARPA contracts FA8750-11-C-0096, FA8750-11-2-0225.

References

1. Abadi, M., Warinschi, B.: Password-Based Encryption Analyzed. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 664–676. Springer, Heidelberg (2005)
2. Baudron, O., Pointcheval, D., Stern, J.: Extended Notions of Security for Multicast Public Key Cryptosystems. In: Montanari, U., Rolim, J.D.P., Welzl, E. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 499–511. Springer, Heidelberg (2000)
3. Bellare, M., Boldyreva, A., Micali, S.: Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (2000)
4. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th FOCS, pp. 394–403. IEEE Computer Society Press (October 1997)
5. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
6. Bellare, M., Ristenpart, T., Tessaro, S.: Multi-instance security and its application to password-based cryptography. Cryptology ePrint Archive, Report 2012/196 (2012), `http://eprint.iacr.org/`

7. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press (November 1993)
8. Boyen, X.: Halting password puzzles: hard-to-break encryption from human-memorable keys. In: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, p. 9. USENIX Association (2007)
9. Boyen, X.: New Paradigms for Password Security (Abstract from the Keynote Lecture). In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. LNCS, vol. 5107, pp. 1–5. Springer, Heidelberg (2008)
10. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)
11. Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård Revisited: How to Construct a Hash Function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)
12. Dodis, Y., Gennaro, R., Håstad, J., Krawczyk, H., Rabin, T.: Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 494–510. Springer, Heidelberg (2004)
13. Dodis, Y., Impagliazzo, R., Jaiswal, R., Kabanets, V.: Security Amplification for *Interactive* Cryptographic Primitives. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 128–145. Springer, Heidelberg (2009)
14. Gennaro, R., Lindell, Y.: A Framework for Password-Based Authenticated Key Exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003)
15. Goldreich, O.: Three XOR-Lemmas — An exposition (1995), `http://www.wisdom.weizmann.ac.il/`
16. Goldreich, O., Impagliazzo, R., Levin, L.A., Venkatesan, R., Zuckerman, D.: Security preserving amplification of hardness. In: 31st FOCS, pp. 318–326. IEEE Computer Society Press (October 1990)
17. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st ACM STOC, pp. 25–32. ACM Press (May 1989)
18. Goldreich, O., Nisan, N., Wigderson, A.: On Yao's XOR-Lemma. In: Goldreich, O. (ed.) Studies in Complexity and Cryptography. LNCS, vol. 6650, pp. 273–301. Springer, Heidelberg (2011)
19. Haitner, I., Harnik, D., Reingold, O.: On the Power of the Randomized Iterate. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 22–40. Springer, Heidelberg (2006)
20. Impagliazzo, R., Kabanets, V.: Constructive Proofs of Concentration Bounds. In: Serna, M., Shaltiel, R., Jansen, K., Rolim, J. (eds.) APPROX and RANDOM 2010, LNCS, vol. 6302, pp. 617–631. Springer, Heidelberg (2010)
21. Kelsey, J., Schneier, B., Hall, C., Wagner, D.: Secure Applications of Low-Entropy Keys. In: Okamoto, E., Davida, G., Mambo, M. (eds.) ISW 1997. LNCS, vol. 1396, pp. 121–134. Springer, Heidelberg (1998)
22. Kohno, T.: Attacking and repairing the winZip encryption scheme. In: Atluri, V., Pfitzmann, B., McDaniel, P. (eds.) ACM CCS 2004, pp. 72–81. ACM Press (October 2004)
23. Krawczyk, H.: Cryptographic Extraction and Key Derivation: The HKDF Scheme. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 631–648. Springer, Heidelberg (2010)
24. Luby, M., Rackoff, C.: A Study of Password Security. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 392–397. Springer, Heidelberg (1988)

25. Maurer, U.M., Pietrzak, K., Renner, R.: Indistinguishability Amplification. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 130–149. Springer, Heidelberg (2007)
26. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
27. Maurer, U., Tessaro, S.: Computational Indistinguishability Amplification: Tight Product Theorems for System Composition. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 355–373. Springer, Heidelberg (2009)
28. Maurer, U., Tessaro, S.: A Hardcore Lemma for Computational Indistinguishability: Security Amplification for Arbitrarily Weak PRGs with Optimal Stretch. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 237–254. Springer, Heidelberg (2010)
29. Morris, R., Thompson, K.: Password security: a case history. Commun. ACM 22, 594–597 (1979)
30. Myers, S.: Efficient amplification of the security of weak pseudo-random function generators. Journal of Cryptology 16(1), 1–24 (2003)
31. Panconesi, A., Srinivasan, A.: Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. SIAM J. Comput. 26(2), 350–368 (1997)
32. PKCS #5: Password-based cryptography standard (rfc 2898). RSA Data Security, Inc., Version 2.0 (September 2000)
33. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with Composition: Limitations of the Indifferentiability Framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 487–506. Springer, Heidelberg (2011)
34. Tessaro, S.: Security Amplification for the Cascade of Arbitrarily Weak PRPs: Tight Bounds via the Interactive Hardcore Lemma. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 37–54. Springer, Heidelberg (2011)
35. Unger, F.: A probabilistic inequality with applications to threshold direct-product theorems. In: 50th FOCS, pp. 221–229. IEEE Computer Society Press (October 2009)
36. Wagner, D., Goldberg, I.: Proofs of Security for the Unix Password Hashing Algorithm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 560–572. Springer, Heidelberg (2000)
37. Yao, A.C.: Theory and applications of trapdoor functions. In: 23rd FOCS, pp. 80–91. IEEE Computer Society Press (November 1982)
38. Yao, F.F., Yin, Y.L.: Design and Analysis of Password-Based Key Derivation Functions. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 245–261. Springer, Heidelberg (2005)