Decoy Password Vaults: At Least as Hard as Steganography?

Cecilia Pasquini, Pascal Schöttle, Rainer Böhme

Decoy Password Vaults: At Least As Hard As Steganography ?

Cecilia Pasquini, Pascal Schöttle, and Rainer Böhme

Department of Computer Science, Universität Innsbruck, Austria
Department of Information Systems, University of Münster, Germany

Abstract. Cracking-resistant password vaults have been recently proposed with the goal of thwarting offline attacks. This requires the generation of synthetic password vaults that are statistically indistinguishable from real ones. In this work, we establish a conceptual link between this problem and steganography, where the stego objects must be undetectable among cover objects. We compare the two frameworks and highlight parallels and differences. Moreover, we transfer results obtained in the steganography literature into the context of decoy generation. Our results include the infeasibility of perfectly secure decoy vaults and the conjecture that secure decoy vaults are at least as hard to construct as secure steganography.

1 Introduction

User-chosen passwords are still the most common authentication standard in online services and users likely cumulate a high number of passwords for different domains. To alleviate the memory effort and possibly let users choose stronger passwords, IT security professionals recommend the use of password vaults (also called "password managers"), which store a user's set of passwords in a container generally encrypted using a single master password.

This encrypted container, stored together with domains and usernames in plaintext, allows users to access websites by just remembering a single password. Furthermore, it can be stored on several (potentially) insecure devices and be backed up in the cloud. Thus, an attacker might get hold of such a container [13, 16] and mount an offline attack against the master password. In comparison to online attacks, which are likely blocked by websites detecting multiple failed login attempts, the effectiveness of an offline attack is only limited by the attacker's computational power. Brute-force attacks are likely successful, as it was shown that human-chosen master passwords have limited entropy and are relatively easy to guess [7, 3].

Although current password-based encryption (PBE) schemes (e. g., PKCS#5 [12]) adopt countermeasures (like the use of a key-derivation function to increase the encryption key entropy, salting to prevent rainbow attacks, or iterative hashing to slow down brute-force attacks), none of these methods can prevent a successful offline attack, as an attacker will always be able to recognize the correctly

decrypted result. In fact, all wrong master password candidates will provide a response that clearly does not resemble user-chosen passwords.

To circumvent this problem, so-called *cracking-resistant password vaults* (CRPVs) have been proposed [2, 5, 10]. The purpose of all CRPVs is to provide an attacker with *honey* or *decoy* vaults even if she decrypts the vault under a wrong master password. These decoy vaults have to be (statistically) indistinguishable from the real vault, so that the real vault is *undetectable* among decoys and the attacker is forced to mount additional online login attempts to identify it.

Another area in information security that shares the protection goal of undetectability is steganography [9]. A *steganographer* wants to communicate a secret message over a communication channel monitored by a *warden* (the attacker in that scenario). The steganographer covertly communicates by modifying a so-called *cover object* (e.g., a digital image) and obtaining a *stego object* that is sent to the intended recipient, and she wants stego objects to be undetectable among cover objects by the warden.

We can summarize the contributions of our paper as follows:

1. we point out the parallels of CRPVs and steganography (Sec. 2);
2. we present a unified model of password vaults and CRPVs (Sec. 3);
3. we transfer established results and security definitions from steganography to the domain of CRPVs, show that perfect security for CRPVs is infeasible and propose the notion of $\varepsilon-$security instead (Sec. 4);
4. we highlight the differences between CRPVs and steganography, conjecturing that secure CRPVs are *at least as hard* to construct as secure steganography (Sec. 5).

Finally, we give an overview of the results obtained and future directions in Section 6.

2 Merging two streams of related work

The already highlighted protection goal of object undetectability represents a clear parallel between CRPVs and steganography, and both communities have made strikingly similar advances.

To overcome security weaknesses of the first CRPV system proposed in [2], the authors of [5] propose the NoCrack system, where decryption under *any* master password yields a plausible decoy vault. The instant creation of decoy vaults is achieved by applying the mechanism of Honey Encryption and Decryption [11]. Despite the name, this approach does not change the encryption/decryption itself, but rather adds another encoding/decoding layer. In particular, a so-called *distribution transformation encoder* (DTE) encodes a plaintext into a bit string and decodes bit strings to plaintexts. The DTE is designed in such a way that random bit strings are decoded to plaintexts following a target statistical distribution, which is hard-coded into the DTE [11]. For instance, an application proposed in [11] is a DTE that mimics the distribution of RSA secret primes

and outputs synthetic primes when decoding a uniform bit string. As we will describe in Section 3.2, a specific DTE is used in the NoCrack system to generate decoy vaults when a wrong master password is used to decrypt the container. A similar approach in steganography has been proposed in 1992, where so-called *mimic functions* [17] are used. Here, Huffman encoding is employed to create text that is statistically indistinguishable from human written text while embedding the secret message. The technique was then extended to arithmetic encoding in model-based steganography [15], where parts of the cover object are replaced by other parts that follow an estimated distribution, similarly to DTEs.

To demonstrate the security of NoCrack, the authors of [5] show that a machine-learning based ranking attack cannot detect the real vault among decoys. A further improvement to the NoCrack system is proposed in the most recent work on CRPVs [10], where the target distribution of the DTE is empirically mixed with the one of the real vault (thus decreasing the statistical difference between real and decoy vaults), and it is also tested against machine-learning classifiers. A relevant similarity to steganography exists, where machine-learning based attacks are used and the results obtained by this are employed to influence "design principles leading to more secure steganography" [8, p. 69].

The NoCrack system [5] with the extension proposed in [10] currently represents the state-of-the-art for CRPVs. In fact, [10] first shows a weakness of the NoCrack system, arguing that the correct vault can be statistically distinguished from the decoys. To achieve this, they use the *Kullback-Leibler divergence* (KLD) between real and decoy vault distribution, which was proposed as an information-theoretical security measure in steganography in 1998 [4].

Due to the high dimensionality of cover and stego objects, steganographers often design their embedding strategies according to *projections* of the whole objects, which are typically simplified models with lower dimensionality [15]. On her side, the warden can employ a different projection that enables her to detect stego objects [9]. This triggered a cat-and-mouse race towards the best projection. In the same way, the DTE in CRPVs reproduces the distribution of a specific projection and the authors of [10] identify the security weaknesses of NoCrack by adopting a different one.

Summarizing, the shared protection goal of undetectability has also led to the use of similar approaches and tools, although, to the best of our knowledge, this link has not been established in the literature yet. This further motivates us to exploit known results in steganography for CRPVs regarding security issues.

3 Password vault model

In this section, we formalize a unified model for CRPV systems. In Section 3.1 we first introduce a general definition for vault objects and identify potential influencing factors. Then, we describe the main components of a CRPV in Section 3.2, focusing on the Honey Encryption and Decryption scheme used.

3.1 Defining password vaults

Password vaults essentially contain credential data. We can formalize credentials as triples (d,u,pw), where d is the domain, u is the username employed and pw is the secret password chosen by the user. Then, a vault $\mathbf{v}$ is a tuple of N credential triples that can be arranged as

$$\mathbf{v} = (\mathrm{d}_1, \ldots, \mathrm{d}_N, \mathrm{u}_1, \ldots, \mathrm{u}_N, \mathrm{pw}_1, \ldots, \mathrm{pw}_N). \tag{1}$$

In practice, $\mathrm{d}_1, \ldots, \mathrm{d}_N$ and $\mathrm{u}_1, \ldots, \mathrm{u}_N$ are plaintext while the vector

$$\mathbf{x} \doteq [\mathrm{pw}_1, \ldots, \mathrm{pw}_N] \tag{2}$$

containing the passwords is encrypted to a ciphertext C under a master password mpw (also user-chosen). We explicitly consider the case where domains and usernames are not encrypted, as in [5, 10], and thus the object to be modeled is given by the vector $\mathbf{x}$. Then, with a slight abuse of notation, in the rest of the paper we will use the term "vault" to indicate only $\mathbf{x}$ instead of the entire tuple $\mathbf{v}$. We can see $\mathbf{x}$ as a realization of a random vector $\mathbf{X}$ with sample space χ^L (χ is the alphabet of symbols used and L is the sum of the N password lengths) and joint probability distribution $\mathcal{P}_{\mathrm{real}}$.

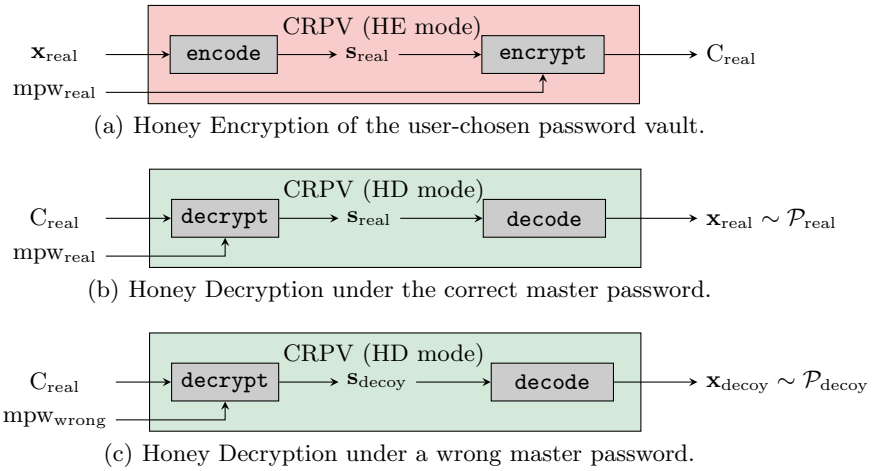
The first part of $\mathbf{v}$, composed by domains and usernames, can have influence on $\mathbf{x}$. It is known that different websites usually adopt specific policies forcing the user to follow certain constraints in choosing the password [2], for instance by requiring a minimum number of symbols, a minimum number of digits and special characters, or the use of both upper- and lower-case letters. Moreover, usernames are often also human-chosen and correlation between the choice of username and password could exist. Thus, the distribution $\mathcal{P}_{\mathrm{real}}$ should be conditioned on the knowledge of domains and usernames, although existing approaches do not always exploit this information. For instance, one of the attacks in [10] specifically uses nonconformity to password policies to successfully detect the real vault among the decoys produced by NoCrack [5].

Even if we discard the dependency on domains and usernames, estimating $\mathcal{P}_{\mathrm{real}}$ is a challenging task, since the statistical behaviour of human-chosen passwords in a vault is highly complex and hard to model. In fact, the partition of $\mathbf{x}$ into independent components (for instance, modeling single password distribution and assuming independence among domains) is highly questionable, as passwords of the same user are typically strongly correlated [2].

Thus, we deal with a joint distribution of L symbols which is hardly observable. However, existing approaches [5, 10] employ a projection $\mathrm{Proj}(\mathbf{x})$ of the entire vector $\mathbf{x}$ and estimate the distribution $\mathcal{P}_{\mathrm{Proj}(\mathbf{X})}$ from available datasets, which is then used to generate synthetic vaults.

3.2 Mimicking vault distribution

CRPVs extend conventional PBE schemes, where a successful or unsuccessful decryption is perfectly recognized, by introducing the use of decoy vaults. We



(a) Honey Encryption of the user-chosen password vault.

(b) Honey Decryption under the correct master password.

(c) Honey Decryption under a wrong master password.

Fig. 1. Honey Encryption and Decryption mechanisms in CRPVs.

now describe how the state-of-the-art CRPV (NoCrack [5]) works and specify which changes are proposed in [10].

NoCrack is a CRPV system that consists of a specific Honey Encryption and Decryption scheme. As introduced in Section 2, the peculiarity of such a mechanism is the use of a DTE, which is a pair of functions $(\texttt{encode}, \texttt{decode})$ with the following properties:

- the input of `encode` is a password vault $\mathbf{x}$ and the output is a binary string $\mathbf{s}$. Conversely, `decode` takes as input any bit string $\mathbf{s}$ and outputs a vault $\mathbf{x}$. It is required that a DTE is *correct*, that is, $\texttt{decode}(\texttt{encode}(\mathbf{x})) = \mathbf{x}$.
- If applied to uniformly distributed bit strings, `decode` should output vaults whose projections follow a known distribution $\mathcal{P}_{\text{Proj}(\mathbf{x})}$.

The authors of [5] devise strategies based on different projections and assumptions, e.g., considering *ℓ-gram* and *Probabilistic Context Free Grammar (PCFG)* models, but the details of DTE design are out of the scope of this section.

The resulting system works as depicted in Fig. 1. For the sake of clarity, we represent the Honey Encryption (HE) and Honey Decryption (HD) modes of the CRPV separately, and for the latter we further distinguish the case of HD with the correct and wrong master password. As mentioned in Section 2, Fig. 1 shows that the DTE $(\texttt{encode}, \texttt{decode})$ is used in combination with a pair of functions $(\texttt{encrypt}, \texttt{decrypt})$, which are based on standard techniques and will not be discussed in detail (we refer the reader to [5] for a thorough description).

When the user chooses the password vault $\mathbf{x}_{\text{real}}$ and the master password mpw_{real}, the HE mode is activated (see Fig. 1(a)). The vault $\mathbf{x}_{\text{real}}$ is processed

by the `encode` function to obtain the string $\mathbf{s}_{\text{real}}$, which is then encrypted under mpw_{real} into a ciphertext C_{real} by means of `encrypt`.[1]

In order to get access to $\mathbf{x}_{\text{real}}$, the user has to decrypt the ciphertext C_{real} by submitting to the system the master password mpw_{real}, thus activating the HD mode. If C_{real} is decrypted under the correct master password mpw_{real}, the user gets as output the real vault $\mathbf{x}_{\text{real}}$ as shown in Fig 1(b).

If an attacker trial-decrypts C_{real} under a wrong master password, `decrypt` outputs a random bit string $\mathbf{s}_{\text{decoy}}$. This string is then given to `decode` that transforms it into a decoy password vault $\mathbf{x}_{\text{decoy}}$, which is delivered to the attacker (see Fig. 1(c)). From her side, the attacker receives a set of password vaults (as many as the number of trial-decryptions), which includes $\mathbf{x}_{\text{real}}$ if and only if mpw_{real} has been used for trial-decrypting.

Regardless of the quality of the algorithm `decode` (i.e., how accurately it transforms random strings into vaults following $\mathcal{P}_{\text{Proj}(\mathbf{X})}$), the use of a DTE will result in a joint distribution $\mathcal{P}_{\text{decoy}}$ of the decoded vaults that is an approximation of $\mathcal{P}_{\text{real}}$. They should be as similar as possible, but the quality of $\mathcal{P}_{\text{decoy}}$ as approximation of $\mathcal{P}_{\text{real}}$ depends on the projection chosen and the database used for the training. In fact, in the attack in [10] the authors identify the correct vault among all the decoys by exploiting a different projection $\mathcal{P}_{\text{Proj}(\mathbf{X})}$ enabling a better distinction. Their improvement then consists exactly in designing the DTE by taking into accounts some statistical properties of the real vault.

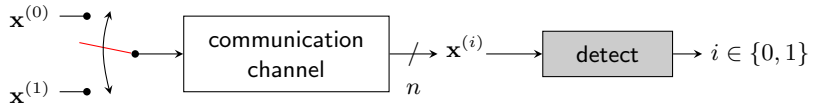
4 Security of CRPV systems

In this section we discuss the security of CRPV systems. First, we compare and translate the definition of perfect security from steganography to the domain of CRPVs in Section 4.1, arguing that this is fundamentally related to the knowledge of the distribution $\mathcal{P}_{\text{real}}$. Then, in Section 4.2 we analyze the computational bounds encountered in studying $\mathcal{P}_{\text{real}}$, giving insights on the practical difficulties in estimating this distribution. Finally, we extend the definition of ε-security to the domain of CRPVs in Section 4.3.

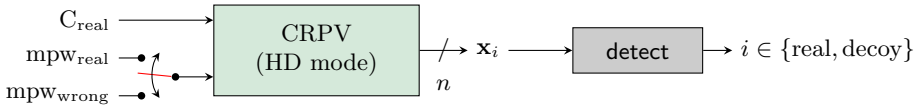
4.1 Perfect security

In steganography, the goal of the steganographer is to transform cover objects $\mathbf{x}^{(0)}$ to stego objects $\mathbf{x}^{(1)}$ containing the secret message, in such a way that the resulting distribution of stego objects $\mathcal{P}_{\text{stego}}$ is close to the distribution of cover objects $\mathcal{P}_{\text{cover}}$. The setup of an attack against a general steganographic system is depicted in Fig. 2(a): depending on the position of the switch (red), cover or stego objects appear on the communication channel. The warden does not control the switch but monitors the channel and applies detect to every object $\mathbf{x}^{(i)}$ that she observes. The output of detect is either 0 indicating that the object

[1] In case of password addition or updating, $\mathbf{x}_{\text{real}}$ is modified and encoded to a new string $\mathbf{s}_{\text{real}}$, which is then encrypted under mpw_{real} to obtain a new ciphertext.



(a) Attack against Steganography



(b) Attack against CRPVs

Fig. 2. Comparison of attacks against steganography and CRPVs

is assumed to be a cover object or 1 if the warden classifies it as stego. The warden wants to identify a secret communication, so her goal is to detect the stego objects among covers.

Attacks against CRPVs can be translated to a very similar setup, as depicted in Fig. 2(b). Here, the attacker has an encrypted password vault $\mathrm{C}_{\mathrm{real}}$ and chooses a set of n master passwords for trial-decryption that might contain the real master password $\mathrm{mpw}_{\mathrm{real}}$ but will be mostly composed of wrong passwords $\mathrm{mpw}_{\mathrm{wrong}}$. Again, the switch (red) indicating whether the chosen password was real or wrong is not under the control of the attacker, although she can decide which and how many master passwords to submit. By this, she ends up with n different password vaults $\mathbf{x}_i$ and she also applies detect to every object $\mathbf{x}_i$ she observes. The output of detect is either real, indicating that $\mathbf{x}_i$ was generated by inputting $\mathrm{mpw}_{\mathrm{real}}$ or decoy, if a $\mathrm{mpw}_{\mathrm{wrong}}$ was chosen. The goal of the attacker is to detect the real vault among the decoys.

Figure 2 opens the way for a formal relationship between $\mathcal{P}_{\mathrm{cover}}$ and $\mathcal{P}_{\mathrm{stego}}$ from steganography and $\mathcal{P}_{\mathrm{real}}$ and $\mathcal{P}_{\mathrm{decoy}}$ in CRPVs. Intuitively, we can view the distribution of real vaults $\mathcal{P}_{\mathrm{real}}$ as the counterpart of the cover distribution $\mathcal{P}_{\mathrm{cover}}$, as both are given by nature and cannot be influenced by either the attacker nor the defender. Both, the distribution of the decoy vaults $\mathcal{P}_{\mathrm{decoy}}$ and the stego distribution $\mathcal{P}_{\mathrm{stego}}$ somehow depend on $\mathcal{P}_{\mathrm{real}}$ and $\mathcal{P}_{\mathrm{cover}}$, respectively.

Based on this analogy, we can recall the definition of perfectly secure steganographic system given in [4] and extend it to CRPVs. According to [4], perfect security in steganography is achieved if and only if the Kullback-Leibler divergence (KLD) between $\mathcal{P}_{\mathrm{cover}}$ and $\mathcal{P}_{\mathrm{stego}}$ is zero, i.e.:

$$\mathrm{KLD}(\mathcal{P}_{\mathrm{cover}} || \mathcal{P}_{\mathrm{stego}}) = 0. \tag{3}$$

Accordingly, perfect security in CRPV systems is achieved iff:

$$\mathrm{KLD}(\mathcal{P}_{\mathrm{real}} || \mathcal{P}_{\mathrm{decoy}}) = 0. \tag{4}$$

In light of that, the action of the detect function in the CRPV domain can be formulated in an hypothesis testing framework, where the null and alternative

hypotheses are given by:

$$\begin{aligned} &H_0\text{: the observed object follows } \mathcal{P}_{\text{real}} \text{ (i.e., it is a real vault)} \\ &H_1\text{: the observed object follows } \mathcal{P}_{\text{decoy}} \text{ (i.e., it is a decoy vault)} \end{aligned} \tag{5}$$

The identification of the real vault is achieved by repeatedly performing the hypothesis test on the n vaults by means of detect. Having zero KLD between two distributions essentially means that they are exactly the same distribution. Thus, with perfect security, the hypotheses in (5) are undecidable.

A fundamental question is whether perfect security in the sense of (4) is possible at all and under which assumptions. In steganography, it is commonly agreed upon by now that this is only possible for so-called *artificial cover sources*, i.e., sources for which the joint distribution $\mathcal{P}_{\text{cover}}$ is fully known, including any conditional dependencies. However, artificial sources do rarely exist in practice. In contrast to that, we deal with *empirical cover sources*, whose distribution is obtained outside the steganographic system from a finite set of observations.

The difference between artificial and empirical cover sources has been proposed in [1], where it is observed that perfect security defined as in (4) generally exists for artificial sources but is impossible for empirical sources. This is related to the fact that in the latter case $\mathcal{P}_{\text{cover}}$ is arguably *incognisable*, and statistical representations by means of proper projections of the sample space will never achieve a zero KLD.

As mentioned in Section 3.2, existing datasets with a finite number of vaults are used to train the DTEs, which then replicate specific statistical properties observed (for instance, ℓ-gram statistics or PCFG statistics). In the next section, we show how hard it is to provide a full characterization of $\mathcal{P}_{\text{real}}$, arguing that $\mathcal{P}_{\text{real}}$ belongs to the class of empirical distributions and is indeed incognisable.

4.2 Computational bounds for the estimation of $\mathcal{P}_{\text{real}}$

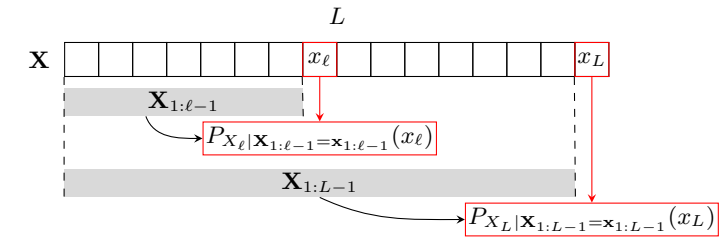
As we introduced in Section 3, in order to fully represent real vaults $\mathbf{x}$, we should consider them as vectors of L symbols regardless of the actual partitioning in different passwords. Thus, each $\mathbf{x}$ is the realization of a L-dimensional discrete random vector $\mathbf{X} = [X_1, \ldots, X_L]$ and the corresponding distribution function $\mathcal{P}_{\text{real}}$ is a joint distribution of L random variables with sample space χ.

Then, $\mathcal{P}_{\text{real}}$ can be expressed by means of the chain rule as follows:

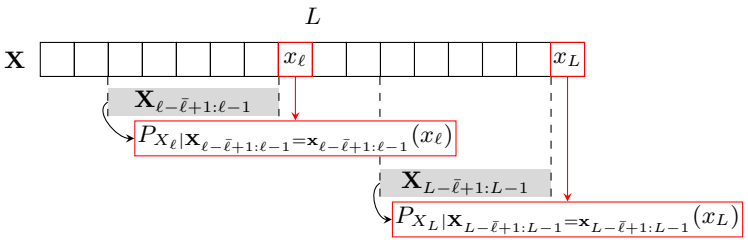
$$\mathcal{P}_{\text{real}}(\mathbf{x}) = \prod_{\ell=1}^{L} P_{X_\ell|\mathbf{X}_{1:\ell-1}=\mathbf{x}_{1:\ell-1}}(x_\ell), \tag{6}$$

where $\mathbf{X}_{i:j}$ is the random vector composed of the random variables (r.v.) in $\mathbf{X}$ from index i to index j and $\mathbf{x}_{i:j}$ is its realization, $P_{X|\mathbf{Y}=\mathbf{y}}(\cdot)$ is the conditional probability mass function (cpmf) of a r.v. X given the realization $\mathbf{y}$ of a random vector $\mathbf{Y}$ and we define $P_{X_1|\mathbf{X}_{1:0}}(x_1) \doteq P_{X_1}(x_1)$.

Let us now suppose to estimate $\mathcal{P}_{\text{real}}$ starting from an available dataset of password vaults, i.e., to approximate each cpmf by means of relative frequencies.



(a) Joint distribution



(b) Markov chain

Fig. 3. Representation of the joint and Markov chain distributions.

In the following, we perform a simple feasibility analysis where we compute the minimum numerosity of the dataset that is necessary to estimate the cpmfs. In doing that, we first consider the joint distribution $\mathcal{P}_{\text{real}}$ in a) and then the case of a specific projection (Markov models) in b) and c):

a) **Joint distribution.** We want to have an approximation of every cpmf in (6). Thus, we need $P_{X_\ell|\mathbf{X}_{1:\ell-1}=\mathbf{x}_{1:\ell-1}}(\cdot)$ for each possible realization $\mathbf{x}_{1:\ell-1}$ of $\mathbf{X}_{1:\ell-1}$, and this holds for $\ell = 1, \dots, L$. A pictorial representation is reported in Fig. 3(a). For each ℓ a number of $|\chi|^{\ell-1}$ cpmfs are then involved. Thus, even assuming that each cpmf is estimated by one single observation (i.e., the support of each cpmf will consist in a single character) the number of necessary vaults is given by $|\chi|^{\ell-1}$. If $\gamma = \log_{10}(|\chi|)$, it is then $\mathcal{O}(10^{\gamma(\ell-1)})$. Let us consider an optimistic setup where a vault contains 10 passwords with average length of 5 characters [2, 7], so that we can reasonably fix $L = 50$. Moreover, let us assume an alphabet corresponding to the printable ASCII characters, thus $|\chi| = 95$ and $\gamma \approx 1.97$. The number of vaults required for the estimation of every $P_{X_{50}|\mathbf{X}_{1:49}=\mathbf{x}_{1:49}}(\cdot)$ with one observation would have a decimal order of magnitude at least equal to $\gamma(50-1) = 96.5$. The number of protons in the universe is estimated by the Eddington number and it is assumed to have on order of magnitude equal to 79. It is worth pointing out that password policies could reduce this number as not the whole alphabet should be used. However, even supposing a restrictive policy where only digits are allowed, this would still result in 10^{49} necessary vaults.

b) **Markov chain of order $\bar{\ell}-1$.** In [5] and [10], DTEs based on $\bar{\ell}$-gram models are trained on an external corpus. As an example, they consider $\bar{\ell}=4$ and estimate the cpmf $P_{X_4|\mathbf{X}_{1:3}=[\mathtt{H},\mathtt{e},\mathtt{l}]}(\mathtt{l})$ as the number of occurrences of the substring "`Hell`" divided by the number of occurrences of the substring "`Hel`" in the corpus. Then, by repeating this for each 3-grams, they estimate a cpmf for each of them and use it to design the DTE.[2]
This is equivalent to considering Markov chains of order $\bar{\ell}-1$, that is, it is assumed that the probability of a character in a certain position depends only on the previous $\bar{\ell}-1$ characters in the vault. As represented in Fig. 3(b), expression (6) is then approximated as

$$\mathcal{P}_{\text{real}}(\mathbf{x}) \approx P_{X_1}(x_1)\cdot \ \ldots \ \cdot P_{X_{\bar{\ell}-1}|\mathbf{X}_{1:\bar{\ell}-2}=\mathbf{x}_{1:\bar{\ell}-2}}(x_{\bar{\ell}-1}) \cdot \prod_{\ell=\bar{\ell}}^{L} P_{X_\ell|\mathbf{X}_{\ell-\bar{\ell}+1:\ell-1}=\mathbf{x}_{\ell-\bar{\ell}+1:\ell-1}}(x_\ell). \tag{7}$$

Again, to obtain each cpmf in expression (7) we would need to observe at least once each possible realization of each $\mathbf{X}_{i:j}$ such that $j-i=\bar{\ell}-1$, and we can again consider a lowerbound of the minimum number of necessary vaults to be $\mathcal{O}(10^{\gamma(\bar{\ell}-1)})$. We report in Table 1 the exponent $\gamma(\bar{\ell}-1)$ as a function of $\bar{\ell}$ in the same setup as before ($|\chi|=95$).

$\bar{\ell}$	2	3	4	5	6	7	8	9	10
$\gamma(\bar{\ell}-1)$	1.97	3.96	5.93	7.91	9.89	11.89	13.84	15.82	17.80

Table 1. Order of magnitude of the minimum number of vaults necessary to the estimation of all the cpmf's when considering a Markov chain of order $\bar{\ell}-1$.

A proper length of the Markov chain is hard to determine and the choice relies on heuristic considerations. However, as we optimistically assumed an average password length equal to 5 symbols, in order to capture the dependencies between different passwords in the vault we should at least consider a Markov chain of order 5 ($\bar{\ell}$=6), so that the probability of the 6-th symbol (likely the first character of the second password) depends also on the realization of the first one. From Table 1, we have that a dataset of at least 7.7 billion ($\approx 10^{9.89}$) vaults would be necessary for this purpose. As of December 2016, the world population was estimated at 7.5 billion, thus implying that the dataset should contain at least one vault for every human being on earth.

c) **Markov model of order $\bar{\ell}-1$ with relaxed assumptions.** We can assume that, for a fixed $\bar{\ell}$, only a fraction p of the cpmfs is actually relevant and

[2] It is to be noted that the authors estimate the cmpfs from datasets of single passwords instead of entire vaults.

the remaining ones can be considered as uniform or estimated via smoothing techniques, as suggested in [5] when building the Markov-based DTE. Then, we can assume that a number of observations $T > 1$ is required for each cpmf in order to have a more accurate approximation. By doing so, the number of vaults is lower bounded by $p \cdot |\chi|^{\bar{\ell}-1} \cdot T$, thus it is $\mathcal{O}(10^{\delta+\gamma(\bar{\ell}-1)})$ where $\delta = \log_{10}(pT)$. The order of magnitude $\delta + \gamma(\bar{\ell} - 1)$ is tabulated in Table 2 for different values of p and T and $\bar{\ell}$ fixed to 6. The values show that even considering as relevant the 0.1% of the realizations of $\mathbf{X}_{1:5}$ and accepting a single observation of each related cpmf ($T = 1$) would require a number of vaults that almost equals the population of Austria (around 8.5 million).

p/T	1	5	10	20
0.001	6.85	7.55	7.85	8.15
0.2	9.15	9.85	10.15	10.45
0.4	9.45	10.15	10.45	10.75
0.6	9.63	10.33	10.63	10.93
0.8	9.75	10.45	10.75	10.05

Table 2. Lowerbound of the minimum number of vaults for the estimation of all the cpmfs when considering a Markov chain of order 5.

In this framework, we should consider that, while the popular RockYou dataset contains more than 32 million passwords in total, the only database of vaults available at the moment (PBvault, see [5]) consists of 276 vaults only. Coupled with the analysis above, which already relies on simplifying assumption like the independence of password from domains and usernames, this strongly motivates our concern on the observability of the full distribution $\mathcal{P}_{\text{real}}$, or even an approximated version of it. So, we can safely say that $\mathcal{P}_{\text{real}}$ is incognisable.

4.3 ε-security

According to our observations in the last subsection, the equality in (4) expressing perfect security is hardly achievable in practice, thus suggesting to consider a non-zero statistical distance between $\mathcal{P}_{\text{real}}$ and $\mathcal{P}_{\text{decoy}}$. In [4], the definition of *ε-security* is introduced, where a system is called ε-secure if

$$\text{KLD}(\mathcal{P}_{\text{real}} || \mathcal{P}_{\text{decoy}}) \leq \varepsilon. \tag{8}$$

If we recall the hypothesis testing framework in (5), we can encounter two different kind of errors:

Type I error: classifying the real vault as a decoy.

Type II error: classifying a decoy vault as the real one.

Denoting with α and β the probabilities of Type I and Type II errors, respectively, inequality (8) is relevant to derive bounds for α and β. With this respect, the Type I error is more relevant than the Type II error for an attacker, since once the real vault is discarded there is no other possibility to successfully obtain the correct passwords. If we accept non-zero KLD, and thus, that α and β cannot be minimized at the same time, we can reasonably think that an attacker would try to achieve $\alpha = 0$ to the detriment of β. It can be shown [14, 4] that, if (8) holds and $\alpha = 0$, the Type II error probability is subject to the lower bound

$$\beta \geq 2^{-\varepsilon}. \tag{9}$$

Inequality (9) provides an interesting link to the required number of online login attempts. In fact, in performing brute force attacks, the attacker will be provided with a set of n vaults, supposedly including the real one. If she enforces $\alpha = 0$ (i. e., the real vault is not misclassified), the number ϕ of plausible candidate vaults identified by `detect` will be approximately at least:

$$\phi = 1 + (n-1)2^{-\varepsilon}. \tag{10}$$

Assuming no further refinements of the candidate selection, $\phi/2$ represents the expected number of online login attempts the attacker is forced to execute. This also addresses an issue that was not explicitly discussed in [10], i.e., the relationship between the ability of detecting the real vault and the total number of decoy vaults. In fact, the authors of [10] consider $n = 1000$ (including the real vault), while an attacker will have to deal with a dramatically higher value of n (equal to the number of trial decryptions) and the performance of the ranking operation in this case is not studied.

5 Differences between steganography and CRPVs

Previous sections concentrated on the similarities of steganography and CRPVs, neglecting obvious differences. In this section, we highlight the main differences and point out their influence on the security of both systems.

(i) **Message embedding.** The most obvious difference between steganography and CRPVs is that in steganography we want to embed a message, which has no direct counterpart in CRPVs. But, in accordance with steganography literature, message embedding can be either seen as a randomization of `encode` or naturally implemented in an adapted version of the DTE. The message encoding problem in steganography is mainly solved, due to the existence of asymptotically perfect codes [6]. So, this difference will not affect the security comparison.

(ii) **Attacker's influence.** Another evident difference is the role of the attacker: in steganography, the warden passively monitors the communication channel and has little influence on the total number n of objects she observes or the relative amount of cover or stego objects. In contrast to that, an attacker

against a CRPV can choose (up to her computational bound) how often she samples $\mathcal{P}_{\text{decoy}}$ and, thus, might refine her model of $\mathcal{P}_{\text{decoy}}$ as accurately as her computational power allows. Even with knowledge of the steganographic algorithm, this is not possible for a warden. Furthermore, with CRPVs the attacker knows that there is at most one real vault. This additional knowledge of an attacker against CRPVs, most likely will have a negative influence on the achievable security of CRPVs.

(iii) **Guessing strategy.** Another degree of freedom that is available to an attacker against CRPVs but not to a steganographic warden is the guessing strategy for the master passwords. If we assume that master passwords are human-chosen, every strategic attacker will choose master passwords in decreasing order of probability, following some model about the *prior* distribution of master passwords $\mathcal{P}_{\text{mpw}}$. For the same arguments explored in Section 4.2, $\mathcal{P}_{\text{mpw}}$ is incognisable. But, the lower dimensionality with respect to $\mathcal{P}_{\text{real}}$ and the higher number of (single) passwords available, e. g., RockYou, would allow for a more accurate estimate of the joint distribution.

(iv) **Oracle queries.** Finally, the possibility of confirming or disproving a vault candidate identified by detect with an online login is probably the highest advantage an attacker against a CRPV has over a warden in steganography. Each online login acts like an oracle query, and the number is only limited by the number of passwords in the vault and the maximum number of wrong login attempts allowed by the different websites. The attacker against a CRPV not only exactly knows when she has the real vault, even negative oracle responses can be used to further refine her estimate of $\mathcal{P}_{\text{decoy}}$ and thus possibly further decreasing β. A warden can only dream of such an oracle in steganography.

Summarizing the above, we conjecture that secure CRPVs are *at least as hard* to construct as secure steganography. Although far away from a formal proof, the existing differences between steganography and CRPVs suggest that the advantage in knowledge an attacker against any CRPV possesses over a warden in steganography will make security of CRPVs ever harder to achieve.

Ultimately, achievable security depends on the evolution of the real distributions. If a cover channel consisting of noise is plausible, then secure steganography reduces to cryptography with the protection goal of indistinguishability of ciphertexts from random sequences. If the users of password vaults choose truly random passwords, constructing secure CRPVs reduces to the generation of random looking sequences. But then, we do not need CRPVs anymore.

6 Conclusion

In this paper we have shown that the parallels between CRPVs and steganography go deeper than the protection goal of undetectability: both fields experienced a similar development, starting from encoding schemes and ending with the employment of machine learning to influence the design of more secure schemes.

While research on CRPVs only started in 2010, the field of digital steganography can look back on more than 25 years of scientific research, thus allowing us to transfer known results to the domain of CRPVs. We believe that leveraging established results in steganography will increase the awareness of researchers when designing new approaches to CRPVs.

Specifically, we argued that the joint distribution of real vaults $\mathcal{P}_{\text{real}}$ is incognisable, due to the data requirements for its full estimation. Even for an approximated version, a dataset containing one password vault for every human being currently living on earth would be needed. The incognisability of $\mathcal{P}_{\text{real}}$ implies that achieving perfect security in CRPVs is *as hard* as constructing perfectly secure steganographic systems in case of empirical sources, thus infeasible in practice. We follow up by arguing that we should rather consider ε-security instead of perfect security. Again, we can leverage established results in steganography and show that we can lower bound the expected amount of online login attempts an attacker is forced to execute when attacking an ε-secure CRPV.

Finally, we conjecture that security in CRPVs is *at least as hard* to achieve as security in steganography due to the differences in both domains' setup. An attacker against a CRPV has several advantages when mounting an attack over a warden in steganography: she can choose the number of trial-decryptions, thus getting a very accurate estimate of the distribution of decoy vaults; she can apply an advanced guessing strategy against the master password, following recent research on how humans choose passwords; and, last but not least, every online login attempt acts as an oracle query, giving the attacker a certain response on whether the vault she faces is the real one or a decoy.

Future work should include formal proofs regarding the effects of the attacker's knowledge on the security of CRPV systems. Moreover, we believe that the conceptual link between steganography and CRPVs is based on the employment of Honey Encryption. Thus, our observations could be extended to other applications of Honey Encryption in practical systems.

Acknowledgments. This research was funded by Deutsche Forschungsgemeinschaft (DFG) under grant "Informationstheoretische Schranken digitaler Bildforensik" and by Archimedes Privatstiftung, Innsbruck, Austria.

References

1. Böhme, R.: An epistemological approach to steganography. In: Katzenbeisser, S., Sadeghi, A.R. (eds.) Information Hiding. Lecture Notes in Computer Science, vol. 5806, pp. 15–30. Springer, Berlin Heidelberg (2009)
2. Bojinov, H., Burszstein, E., Boyen, X., Boneh, D.: Kamouflage: loss-resistant password management. In: European Symposium on Research in Computer Security (ESORICS). pp. 286–302 (2010)
3. Bonneau, J.: Guessing human-chosen secrets. Ph.D. thesis, University of Cambridge (May 2012)
4. Cachin, C.: An information-theoretic model for steganography. In: Aucsmith, D. (ed.) Information Hiding, Lecture Notes in Computer Science, vol. 1525, pp. 306–318. Springer, Berlin Heidelberg (1998)

5. Chatterjee, R., Bonneau, J., Juels, A., Ristenpart, T.: Cracking-resistant password vaults using natural language encoders. IEEE Security and Privacy pp. 481–498 (2016)
6. Filler, T., Judas, J., Fridrich, J.: Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Transactions on Information Forensics and Security 6(3), 920–935 (2011)
7. Florencio, D., Herley, C.: A large-scale study of web password habits. In: ACM International Conference on World Wide Web. pp. 656–666 (2007)
8. Fridrich, J.: Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In: Information Hiding. pp. 67–81. Springer, Berlin Heidelberg (2005)
9. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, New York, NY, USA (2009)
10. Golla, M., Beuscher, B., Dürmuth, M.: On the security of cracking-resistant password vaults. In: ACM Conference on Computer and Communications Security. pp. 1230–1241 (2016)
11. Juels, A., Ristenpart, T.: Honey encryption: beyond the brute-force barriers. In: Advances in Cryptology - EUROCRYPT. pp. 293–310 (2014)
12. Kaliski, B.: PKCS# 5: Password-based cryptography specification version 2.0. RFC 2289 (2000)
13. Li, Z., He, W., Akhawe, D., Song, D.: The emperor's new password manager: security analysis of web-based password managers. In: USENIX Security Symposium. pp. 465–479 (2014)
14. Maurer, U.M.: A unified and generalized treatment of authentication theory. In: Puech, C., Reischuk, R. (eds.) 13th Annual Symposium on Theoretical Aspects of Computer Science. pp. 387–398. Springer, Berlin, Heidelberg (1996)
15. Sallee, P.: Model-based steganography. In: Kalker, T., Cox, I., Ro, Y. (eds.) Digital Watermarking, Lecture Notes in Computer Science, vol. 2939, pp. 254–260. Springer Berlin Heidelberg (2004)
16. Silver, D., Jana, S., Boneh, D., Chen, E.: Password managers: attacks and defenses. In: USENIX Security Symposium. pp. 449–464 (2014)
17. Wayner, P.: Mimic functions. Cryptologia 16(3), 193–214 (Jul 1992)