Towards the Quantification of Strategy Leakage

Mário S. Alvim*, Piotr Mardziel†, Michael Hicks‡

*Universidade Federal de Minas Gerais msalvim@dcc.ufmg.br
†Carnegie Mellon University piotrm@cmu.edu
‡University of Maryland, College Park mwh@cs.umd.edu

***Abstract*—This paper reports first steps towards a formal model for *strategy leakage*. We generalize the representation of prior adversarial knowledge from a distribution on secrets to a distribution on *strategies* for generating secrets, which we call an *environment*. Applying information-theoretic techniques to environments allows us to disentangle the information leakage about a secret from the leakage about how users generate their secrets, i.e., their *strategy*. Such an approach can provide insight in situations when secrets change according to a strategy (e.g., location based on commuting patterns), or when multiple secrets are generated according to a strategy (e.g., passwords on different web sites), among others.**

***Index Terms*—quantitative information flow, strategy leakage, formal methods.**

I. Introduction

Two core principles within the field of *quantitative information flow* (QIF) are: (i) a secret is considered "safe" to the extent the probability distribution on secret values has high entropy; and (ii) the leakage of information in a system is a measure of how much the observable behavior of the system, while processing a secret value, degrades the entropy of that secret. These principles have been used to create ever more sophisticated QIF frameworks to model systems and reason about leakage. However, little attention has been paid to understand the sources that inject entropy into the distribution on secrets in the first place.

Consider the example of passwords. Users do not generate (completely) random passwords. Rather, they generate them according to a *strategy* (which might involve some randomness). A person born in 1983, for example, might have a strategy of generally picking passwords containing the string "1983". When systems mandate regular password changes, the actual benefit obtained will depend on the strategies users employ to generate new passwords. Indeed, studies have shown that expiration policies may not be as beneficial as believed [10, 2, 8].

The question we are interested in is the following: **To what extent does knowledge of *strategies* help the adversary guess the secret?** To be able to answer this question, we develop a model that not only considers the space of possible secrets, but also considers the space of possible strategies that generated them.

Returning to our password example, suppose an adversary gains access to a large collection of passwords (without the associated user identities). We can derive from such a database a model of adversary prior knowledge: a probability distribution over passwords. This model makes no assumptions about how these passwords were chosen: effectively, the prior aggregates a population of users into a single probabilistic behavior. But suppose that the adversary knew that passwords with "1983" in them were more likely to have been generated by people born in 1983. Then on a system that mandates password changes, the adversary may have an advantage when guessing that a changed password by the same user has the same birth year substring.

Generally speaking, knowledge of a secret-generating strategy can be useful when that strategy is used to produce multiple secrets. For example, the same user might use a similar strategy to generate passwords on different web sites. As another example, a new secret may be correlated with an old secret, according to the strategy used. Aside from passwords, if we consider locations as secret, then changes in location are surely correlated, e.g., based on time of day. Perhaps surprisingly, an evolving secret subject to repeated observations, in some cases, can be learned *faster* if it is changed (and observed) more often [3]. The reason is that the strategy by which the secret changes is revealed faster if more samples from the strategy are visible to an adversary; and if the strategy has little randomness in it, the adversary has an increased accuracy in determining past, current, and even future secret values.

Modeling both secrets and secret-generating strategies lets us consider adversaries being able to learn the strategy, and ideally, quantify the advantage this provides them. Such an advantage would thus constitute the value in knowing the strategy as distinct from knowing the secret. In the rest of this short paper we present our initial model and characterizations of strategy leakage, as well as a small numerical demonstration based on a real leaked password dataset.

II. Preliminaries

We briefly review fundamental concepts and notation from *quantitative information flow* (QIF). Notably we define notions of secret, an adversary's uncertainty about the secret, a measure assessing that uncertainty, and (in anticipation of our notion of *environment*) uncertainty (and a measure thereof) over strategies that generate the secrets.

An adversary usually only has partial information about the value of a *secret*, modeled by a *prior* probability distribution.

Object	Type	Typical instance
secret	$\mathcal{X}$	x
prior distribution	$\mathbb{D}\mathcal{X}$	p_X
prior vulnerability (of the secret)	$\mathbb{D}\mathcal{X} \to \mathbb{R}$	$\mathbb{V}$
strategy	$\mathcal{S} = \mathbb{D}\mathcal{X}$	s
environment	$\mathbb{D}\mathcal{S} = \mathbb{D}^2\mathcal{X}$	p_S
contextual vulnerability (of the secret)	$\mathbb{D}\mathcal{S} \to \mathbb{R}$	$\mathbb{V}_C$
vulnerability of the strategy	$\mathbb{D}\mathcal{S} \to \mathbb{R}$	$\mathbb{V}_S$

TABLE I
NOTATION.

We denote by $\mathcal{X}$ the set of possible secrets and by $\mathbb{D}\mathcal{X}$ the set of probability distributions over $\mathcal{X}$. We typically use p_X to denote a prior.

An information measure is a function $\mathbb{V} : \mathbb{D}\mathcal{X} \to \mathbb{R}$ mapping priors to real numbers. An information measure can gauge prior *vulnerability* – the higher the value, the less secure the secret is – or prior *uncertainty/entropy* – the higher the value, the more secure the secret is. There are several definitions of information measures in the literature, varying according to the operational interpretation of the measure. Popular instances include *Bayes-vulnerability* [7], *Shannon-entropy* [6], and *guessing-entropy* [4]. Recently, the more general *g-vulnerability* framework [1] proposed vulnerabilities to capture many different adversarial models. In the remainder of this paper we will use the term "vulnerability" for measures in general though our claims also apply to uncertainty measures.

A *hyper-distribution* [5] (or *hyper* for short) is a distribution on distributions. A hyper on the set $\mathcal{X}$ is of type $\mathbb{D}^2\mathcal{X}$, which stands for $\mathbb{D}(\mathbb{D}\mathcal{X})$, a distribution on distributions on $\mathcal{X}$. The elements of $\mathbb{D}\mathcal{X}$ are called the *inner-distributions* (or *inners*) of the hyper. The distribution the hyper has on inners is called the *outer-distribution* (or *outer*). We use $\lceil\Delta\rceil$ to denote the *support* (the set of inners with non-zero probability) of hyper Δ, and $[p_X]$ to denote the point-hyper assigning probability 1 to the inner p_X.

We denote the *expected value* of some random variable $F : \mathcal{X} \to R$ over a distribution $p_X : \mathbb{D}\mathcal{X}$ by

$$\mathop{\mathbb{E}}_{p_X} F \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{x \leftarrow p_X} F(x) \stackrel{\text{def}}{=} \sum_{x \in \mathcal{X}} p_X(x) F(x) \ .$$

Here, R is usually the reals $\mathbb{R}$ but more generally can be a vector space. If $\mathcal{X}$ itself is a vector space, then we abbreviate $\mathbb{E}_{p_X}(\mathrm{id})$ by just $\mathbb{E}\, p_X$, the "average" of the distribution p_X on $\mathcal{X}$.

III. STRATEGIES AND THEIR VULNERABILITY

This section presents our model which makes explicit that secrets are generated by applying a particular strategy. We develop measures both for the vulnerability of the secret when its possible generating strategies are considered (called contextual vulnerability), and the vulnerability of the particular strategy that generates the secret. We observe that for QIF uncertainty measures, the traditional definition of the measure decomposes neatly into contextual vulnerability and strategy vulnerability.

A. Contextual vulnerability of the secret

A *strategy* is a probabilistic rule for generating secrets, represented by a probability distribution on the set $\mathcal{X}$ of secrets. The set $\mathcal{S}$ of all strategies is thus $\mathbb{D}\mathcal{X}$.

We model the *environment* in which the secret is generated as a probability distribution p_S on the set $\mathcal{S}$ of strategies. The set $\mathbb{D}\mathcal{S}$ of all environments is the set of all probability distributions on strategies, i.e., the set $\mathbb{D}^2\mathcal{X}$ of hypers on $\mathcal{X}$. The environment represents the actual space from which strategies are sampled.

We assume the prior p_X on secrets is *consistent* with the environment p_S, that is, that the prior is obtained as the expected behavior of the environment lacking any knowledge about strategy used:

$$p_X \ = \ \mathop{\mathbb{E}} p_S \ . \tag{1}$$

The concept of environment can capture the interpretation of a prior used in many traditional approaches to QIF: in the case there is a single strategy p_X to generate secrets, the environment p_S is the point-hyper $[p_X]$.

Whereas in traditional models of QIF the adversary only knows the prior p_X on secrets, in the following we will also consider a more powerful type of adversary, one who knows the strategy p_S being used. The definitions that follow will be phrased in terms of the expectation over all the possible strategies p_S.

Definition 1. We say an adversary *knows the environment* if he has full knowledge of the hyper-distribution p_S and can tell what strategy is being used to generate the secret.

The vulnerability of the secret for this type of more powerful adversary [1] is defined as follows.

Definition 2. Given an information measure $\mathbb{V}$ on secrets, the *contextual vulnerability of the secret* (or *contextual vulnerability*, for short) is a function $\mathbb{V}_C : \mathbb{D}^2\mathcal{X} \to \mathbb{R}$ defined as

$$\mathbb{V}_C(p_S) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{p_S} \mathbb{V} \ ,$$

representing the expected vulnerability of the secret if the environment (i.e., the context) is known to the adversary,.

A first result is that in expectation an adversary who knows the environment cannot be worse off than an adversary who only knows the prior distribution on secrets.

[1] The modeling of adversaries with intermediate levels of knowledge about the strategy and/or secret are the object of current investigation.

Proposition 1. *If $\mathbb{V}$ is a g-vulnerability (or, equivalently, any continuous, convex function), then $\mathbb{V}_C(p_S) \geq \mathbb{V}(p_X)$.*

Moreover, in case the environment p_S is a point-hyper $[p_X]$, contextual vulnerability $\mathbb{V}_C(p_S)$ collapses into $\mathbb{V}(p_X)$.

Proposition 2. *If $p_S = [p_X]$, $\mathbb{V}_C(p_S) = \mathbb{V}(p_X)$.*

The converse of Proposition 2 is not true, i.e., $\mathbb{V}_C(p_S) = \mathbb{V}(p_X)$ does not imply $p_S = [p_X]$. This observation is at the heart of the definition of strategy vulnerability, which we will discuss in Section III-C.

B. Real security vs. security from aggregation

The modeling of adversarial knowledge as only a prior on secrets p_X overlooks how the adversary can exploit knowledge of the environment p_S to uncover secrets. We demonstrate this fact with an example that shows that secrets distributed according to the same prior may present drastically different contextual vulnerability.

Define a set $\mathcal{X} = \{x_1, x_2\}$ of binary secret values, and a set $\mathcal{S} = \{s_1, s_2, s_3\}$ of possible strategies, where

- $s_1 = [1, 0]$ always generates value x_1,
- $s_2 = [0, 1]$ always generates value x_2, and
- $s_3 = [1/2, 1/2]$ generates each value with equal probability.

We define the two environments:

- $p_S^1 = [1/2, 1/2, 0]$ is the environment in which strategies s_1 and s_2 may be adopted with equal probability, and
- $p_S^2 = [0, 0, 1]$ is the environment in which strategy s_3 is always adopted.

We depict strategies and environments in the following table. The columns list strategies; the first grouping of rows contains the definition of the strategy (i.e., the probability that it chooses a particular secret), and the next grouping of rows contains the definition of each environment, one per row, which gives the probability of each strategy.

	s_1	s_2	s_3
x_1	1	0	$1/2$
x_2	0	1	$1/2$
p_S^1	$1/2$	$1/2$	0
p_S^2	0	0	1

Both environments yield the same prior distribution $p_X = \mathrm{E}\, p_S^1 = \mathrm{E}\, p_S^2 = [1/2, 1/2]$, so an adversary who cannot observe what strategy is being used would obtain the same prior vulnerability in both environments. For instance, for Bayes-vulnerability $\mathbb{V}^{\text{Bayes}}(p_X) \stackrel{\text{def}}{=} \max_{x \in \mathcal{X}} p_X(x)$, the adversary would obtain a prior vulnerability $\mathbb{V}^{\text{Bayes}}(p_X) = 1/2$.

However, an adversary who can learn the strategy being used will obtain different values for the vulnerability of the secret in each environment. In environment p_S^1 the contextual vulnerability is

$$\begin{aligned}\mathbb{V}_C^{\text{Bayes}}(p_S^1) &= 1/2 \cdot \mathbb{V}^{\text{Bayes}}(s_1) + 1/2 \cdot \mathbb{V}^{\text{Bayes}}(s_2) \\ &= 1/2 \cdot 1 + 1/2 \cdot 1 = 1,\end{aligned}$$

whereas in environment p_S^2 the contextual vulnerability is

$$\mathbb{V}_C^{\text{Bayes}}(p_S^2) = 1 \cdot p(s_3) = 1 \cdot 1/2 = 1/2.$$

Note that in environment p_S^2, the value for contextual vulnerability and prior vulnerability is the same ($\mathbb{V}_C^{\text{Bayes}}(p_{S^2}) = \mathbb{V}^{\text{Bayes}}(p_X) = 1/2$), so an adversary who learns the strategy being used is not expected to be more successful than an adversary who only knows the prior.

On the other hand, in environment p_{S^1}, contextual vulnerability exceeds prior vulnerability ($\mathbb{V}_C^{\text{Bayes}}(p_{S^1}) = 1 > 1/2 = \mathbb{V}^{\text{Bayes}}(p_X)$), and an adversary who learns the exact strategy being used is expected to be successful twice as often as an adversary who only knows the prior.

We can take two things from this example. First, *security from aggregation* occurs when contextual vulnerability exceeds prior vulnerability: $\mathbb{V}_C(p_S) \gg \mathbb{V}(p_X)$. In this case the secret is protected by the adversary's lack of knowledge of the strategy being used, and, if the adversary learns the strategy, the vulnerability of the secret can significantly increase. An example of security from aggregation is a scenario in which all users pick passwords with deterministic strategies, but the adversary does not know which user is generating the password. If there is a large number of users, and if their strategies are varied enough, the passwords may be considered "secure" only as long as the adversary cannot use knowledge about the environment to identify the strategy being used.

On the other hand, *real security* occurs when contextual and prior vulnerabilities have similar values: $\mathbb{V}_C(p_S) \approx \mathbb{V}(p_X)$. In this case the secret is protected by the unpredictability (or uncertainty) within the strategies that generate the secret. In this case, even if the strategy becomes known to the adversary, the vulnerability of the secret will not increase significantly. An example of real security is a bank system in PINs are chosen for each user uniformly. Even if the algorithm is known to the adversary, the vulnerability of the secret is not increased.

C. Strategy vulnerability

We now turn our attention to how the knowledge of an environment reflects on the adversary's knowledge about the strategy being used to generate secrets. Here we are concerned with how certain an adversary is about which strategy is being used, independently of whether the strategy itself is deterministic or random. We want to be able to characterize, in particular, environments in which the adversary knows exactly the strategy being used, but that strategy just happens to be random (e.g., a uniform distribution over secret values), and environments in which the adversary does not know what strategy is being used, even if all possible strategies are deterministic. For that we will define a measure $\mathbb{V}_S(p_S) : \mathbb{D}\mathcal{S} \to \mathbb{R}$ of *strategy vulnerability*.

The key insight underlying our definition is that $\mathbb{V}_S(p_S)$ should consider the "similarity" among strategies in the support of p_S. From the point of view of the adversary, whose goal is to guess the secret, two strategies should be considered "similar" if they yield "similar" vulnerabilities of the secret, as measured according to some $\mathbb{V} : \mathbb{D}\mathcal{X} \to \mathbb{R}$ of interest.

As such, the vulnerability of a particular environment should be higher when its support consists of similar strategies, and lower when strategies are very different.

To understand this, consider an extension from our prior example, adding a strategy s_4 and environment p_S^3:

	s_1	s_2	s_3	s_4
x_1	1	0	$1/2$	0.9
x_2	0	1	$1/2$	0.1
p_S^1	$1/2$	$1/2$	0	0
p_S^2	0	0	1	0
p_S^3	$1/2$	0	0	$1/2$

Intuitively, the strategy vulnerability in p_S^2 should be high: the adversary knows exactly the strategy being used. What should be the strategy vulnerability in p_S^1 and in p_S^3? Suppose we defined $\mathbb{V}_S$ as a kind of Bayes vulnerability:

$$\mathbb{V}_S^{\text{1st}}(p_S) \stackrel{\text{def?}}{=} \max_{s\in\mathcal{S}} p_S(s)\ . \tag{2}$$

Then we have $\mathbb{V}_S^{\text{1st}}(p_S^1) = 1/2$, $\mathbb{V}_S^{\text{1st}}(p_S^2) = 1$, and $\mathbb{V}_S^{\text{1st}}(p_S^3) = 1/2$. But this seems wrong. We are assigning the same measure of vulnerability to both p_S^1 and p_S^3, but these two environments are very different. The possible strategies in environment p_S^1 never produce the same secret. The strategies of p_S^3, on the other hand, produce secrets x_1 and x_2 with similar probabilities. $\mathbb{V}_S^{\text{1st}}$ ascribes p_S^1 and p_S^3 the same measure even though the vulnerability of the secret under knowledge of p_S^3 is much higher than p_S^1. That is, an adversary who knows p_S^3 would always guess the secret to be x_1 and would be right most of the time, but an adversary who knows p_S^1 gains no advantage about which secret to guess. In short, we want $\mathbb{V}_S(p_S^2) > \mathbb{V}_S(p_S^3) > \mathbb{V}_S(p_S^1)$, but $\mathbb{V}_S^{\text{1st}}$ fails to satisfy this ordering.

These observations lead us to define the vulnerability of a strategy in terms of the *difference in accuracy*, as measured by a choice of $\mathbb{V}$, of an adversary betting according to its full knowledge of the environment p_S and an adversary betting according to the expected behavior of the environment, that is, according to the prior defined in Equation (1) as $p_X = \mathrm{E}\, p_S$. A strategy is, *for practical purposes*, known to the adversary when $\mathbb{V}(p_X) \approx \mathbb{V}_C(p_S)$, or, equivalently, when $\mathbb{V}(\mathrm{E}\, p_S) \approx \mathrm{E}_{p_S}\, \mathbb{V}$.

Definition 3. If $\mathbb{V}$ is a vulnerability measure (such as Bayes vulnerability or g-vulnerability), then *strategy vulnerability* is the ratio $\mathbb{V}_S(p_S) \stackrel{\text{def}}{=} \mathbb{V}(\mathrm{E}\, p_S)/\mathbb{V}_C(p_S)$. The ratio is inverted for uncertainty measures (such as guessing entropy).

Note that we have always $0 \le \mathbb{V}_S(p_S) \le 1$ (by Proposition 1), and $\mathbb{V}_S(p_S)$ is maximum when $\mathbb{V}(\mathrm{E}\, p_S) = \mathbb{V}_C(p_S)$. Moreover, note the definition is consistent with the decomposition of prior vulnerability into the product of strategy vulnerability and contextual vulnerability: $\mathbb{V}(p_X) = \mathbb{V}_S(p_S)\cdot\mathbb{V}_C(p_S)$.

IV. Case Study

To illustrate the utility of our model, we synthesize environments based on the RockYou password dataset [9], which contains the un-hashed passwords of around 32 million users of the RockYou gaming site. For each environment we compute the corresponding contextual vulnerability and strategy vulnerability, showing how they relate.

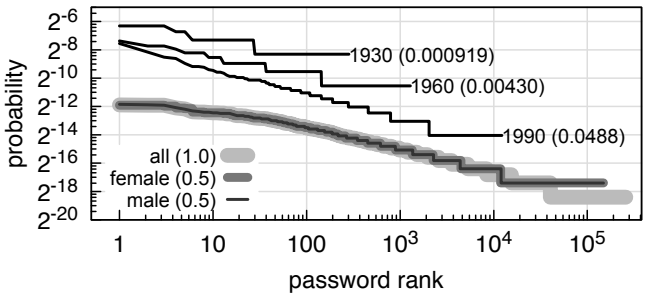

Fig. 1. Example strategies and their probabilities in several environments: "all", the sole strategy in the **Prior** environment, the "male" and "female" strategies in the **Gender** environment, and 3 age-aggregate strategies in the **Age** environment.

We begin by reducing the 32 million passwords to the around 350k passwords that contain a string suggesting the birth year of the password's owner (the strings "1917" through "1995"). We attributed to each password the synthesized birth year as well as a randomly chosen gender.

To synthesize our first environment, which we call **Omniscient**, we construct a deterministic strategy for each of the 350k passwords in the reduced database. Each strategy represents a user along with their exact preference at the time they selected their password. This level of knowledge is beyond the reach of any adversary but will serve to illustrate the edge cases of our calculations.

To construct the **Age** environment, we partition the passwords into blocks according to the birth year attributed to that password. From each block we derive a distribution on passwords representing the strategy for a person born in that year. This produces one strategy for each birth year from 1917 through 1995. The probability of each strategy is determined by the relative frequency of each birth year.

The **Gender** environment aggregates strategies into one male and one female strategy. As we assigned genders to passwords uniformly at random, these two strategies each occur with equal 0.5 probability and are mostly similar.

Finally, the **Prior** environment has only one strategy in its support that aggregates all of the 350k passwords, with each password's probability being proportional to its relative frequency. This environment is equivalent to the point hyper $[p_X]$ containing only the prior distribution on secrets.

Several strategies in the last three environment are visualized in Figure 1. The "all" line shows the probability of various passwords being picked in the **Prior** environment, sorted by their rank (most probable first). The two gender aggregate strategies from the **Gender** environment are labeled "male" and "female" (note that "male", "female" and "all" largely coincide). Finally, three example years from the **Age**

	$\mathbb{V}^{Bayes}(p_X)$	$=$	$\mathbb{V}_C^{Bayes}(p_S)$	$*$	$\mathbb{V}_S^{Bayes}(p_S)$
Omni	$2^{-11.892}$	$=$	2^{-0}	$*$	$2^{-11.892}$
Age	$2^{-11.892}$	$=$	$2^{-7.442}$	$*$	$2^{-4.450}$
Gender	$2^{-11.892}$	$=$	$2^{-11.876}$	$*$	$2^{-0.0158}$
Prior	$2^{-11.892}$	$=$	$2^{-11.892}$	$*$	2^{-0}

TABLE II
BAYES VULNERABILITY DECOMPOSITION

environment are labeled "1930", "1960", and "1990". The Bayes vulnerability of each strategy is the probability of the rank 1 password and min-entropy is negation of the base 2 exponent of that probability.

The decomposition of prior Bayes vulnerability as per Definition 3 is summarized in Table IV. We can see there that the vulnerability in the prior is around $2^{-11.892} = 2.632 \cdot 10^{-4}$ (or equivalently 11.892 bits of min-entropy). In the environment where gender is learnable by the adversary, they would achieve vulnerability of $2^{-11.876} = 2.66084 \cdot 10^{-4}$ were they to learn the target user's gender. The strategy vulnerability in this case shows a negligible advantage over the prior because we synthesized the gender uniformly in this scenario. On the other hand, in the case of the age aggregated environment, learning the age would result with the vulnerability $2^{-7.442} = 57.526 \cdot 10^{-4}$, providing the equivalent of 4.450 bits of information over the prior when measured as min-entropy.

V. CONCLUSION

This paper has presented generalized measures of vulnerability and/or uncertainty about a secret to measures over the possible strategies employed to generate the secret. This generalization permits reasoning about the "accelerated strategy leakage" phenomenon in which frequently changing secrets can more quickly reveal their strategy for change, ultimately making current/future secret values more vulnerable. We are currently working on characterizations of intermediate levels of adversarial knowledge about the environment, and on how to apply the tools to measure leakage of a channel to the scenarios where adversarial knowledge is more refined than a prior.

Acknowledgments: Research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. This work was developed with the support of Conselho Nacional de Desenvolvimento Cientfico e Tecnolgico (CNPq), and Fundao de Amparo Pesquisa do Estado de Minas Gerais (FAPEMIG).

REFERENCES

[1] Mário S. Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. "Measuring Information Leakage Using Generalized Gain Functions". In: *Proceedings of the IEEE Computer Security Foundations Symposium (CSF)*. 2012.

[2] Sonia Chiasson and Paul C van Oorschot. "Quantifying the security advantage of password expiration policies". In: *Journal of Designs, Codes, and Cryptography* 77.2-3 (2015), pp. 401–408.

[3] Piotr Mardziel, Mário S. Alvim, Michael Hicks, and Michael Clarkson. "Quantifying Information Flow for Dynamic Secrets". In: *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*. May 2014.

[4] James L. Massey. "Guessing and Entropy". In: *Proceedings of the IEEEInternational Symposium on Information Theory (ISIT)*. IEEE, 1994, p. 204.

[5] Annabelle McIver, Larissa Meinicke, and Carroll Morgan. "Compositional Closure for Bayes Risk in Probabilistic Noninterference". In: *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*. 2014.

[6] Claude Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27 (1948), pp. 379–423, 623–656.

[7] Geoffrey Smith. "On the Foundations of Quantitative Information Flow". In: *Proceedings of the Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*. 2009.

[8] *Time to rethink mandatory password changes*. https://www.ftc.gov/news-events/blogs/techftc/2016/03/time-rethink-mandatory-password-changes. Accessed: 2016-04-15.

[9] Ashlee Vance. *If Your Password Is 123456, Just Make It HackMe*. http://www.nytimes.com/2010/01/21/technology/21password.html. Accessed: 2016-04-16.

[10] Yinqian Zhang, Fabian Monrose, and Michael K Reiter. "The security of modern password expiration: an algorithmic framework and empirical analysis". In: *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. 2010.