A New Multimodal Approach for Password Strength Estimation—Part II: Experimental Evaluation

Javier Galbally, Iwen Coisel, and Ignacio Sanchez

***Abstract*— A novel multimodal method for the estimation of password strength was presented in Part I of this series of two papers. In this paper, the experimental framework used for the evaluation of the novel approach is described. The method is evaluated following a reproducible protocol, which includes a three-dimensional approach: 1) deterministic assessment; 2) statistical assessment; and 3) third parties assessment (thanks to the availability upon request of an executable application that integrates the multimodal meter). The key experiment of the protocol compares, from a probabilistic point of view, the strength distributions assigned to passwords broken with increasingly complex attacking approaches, following a common strategy in a typical password cracking session. The experimental evaluation is carried out not only for the new meter, but also for other strength estimators from the state of the art, comparing their overall performance. In addition to its consistent results, the proposed method is highly flexible and can be adjusted to specific environments or to a certain password policy. Furthermore, it can also evolve over time in order to naturally adjust to new password selection trends followed by users.**

***Index Terms*— Password security, strength meters, password evaluation, multimodality, password policies.**

I. Introduction

IN PART I of this series of two papers [1], we introduced the theoretical framework of a novel multimodal method for the estimation of password strength. The new meter presents two key by-design characteristics:

- **Multimodality**. The main rationale behind the development of a multimodal approach to evaluate password strength is that: by exploiting the advantages of different individual techniques through their fusion, it will be possible to achieve one unique multimodal measure which overcomes many of their weaknesses.
- **Flexibility**. Password strength estimation algorithms should not be immutable. On the contrary, they should be able to adapt to different application-specific environments depending on the language used, hashing algorithm, alphabet, etc. Following this principle, the individual modules that conform the overall multimodal method have been developed to be flexible. They present a number of parameters that should be fixed on a case by case basis during an initial training phase. This way, the method can be adapted to provide more accurate strength estimations for each particular scenario (e.g., English based application *VS* Russian based application).

The present Part II of this series of two works, focuses on: 1) the description of the experimental framework followed to evaluate the novel strength estimator presented in Part I; 2) the analysis of the results obtained compared to other state of the art methods.

One of the main challenges to be faced in the development of password strength meters is the assessment of their performance. Unlike other problems related to the field of machine-learning, in this case there is no ground-truth data with which to compare the results of new meters, as the strength of a password is an intrinsically subjective value. For the sake of argument, let's assume two different strength meters that assign to password "Pet52!" a score of 2 and 4 respectively, both in a scale from 0 to 10. The question to be addressed is: which of the two is more accurate? There is not a unique valid answer to that question since there is not a "universally correct" strength value that can serve as validation measure. However, even if a fully objective evaluation seems difficult, common sense dictates that if a password like "maria" is given a higher strength than "Swy6oi28rE?!Hf", the corresponding meter is not a good estimator.

Given the difficulties posed by the lack of ground-truth data for the objective assessment of password strength meters, there is still no standard methodology on how the problem should be approached. In the literature, just a few works have addressed the challenge of comparing the performance of several strength estimators in order to determine their strengths and shortcomings [2]–[8].

In the present paper we build upon the lessons learned from those previous valuable works, in order to present a new full evaluation protocol inspired in the principles used for the assessment of algorithms related to machine-learning (such as the Markov Chains). This way, the new multimodal method is trained and tested on different datasets, both from a: 1) Statistical perspective: to determine the correlation between the robustness of passwords to attacks of increasing complexity and the strength assigned by different meters; 2) Deterministic perspective: to establish the consistency of strength estimation methods on specific passwords. The evaluation protocol includes a new overall assessment score for strength meters that allows comparing in a fast and quantitative manner different algorithms.

Following the previous discussion, the contributions of this Part II may be summarized as follows: 1) the new assessment protocol followed to assess the proposed technique. It has

Manuscript received January 11, 2017; revised May 12, 2017 and June 16, 2017; accepted July 8, 2017. Date of publication July 21, 2017; date of current version August 29, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Karen Renaud. *(Corresponding author: Javier Galbally.)*

The authors are with the European Commission, Joint Research Center (DG JRC), 21027 Ispra, Italy (e-mail: javier.galbally@ec.europa.eu; iwen.coisel@ec.europa.eu; ignacio.sanchez@ec.europa.eu).

This paper has supplementary downloadable material at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2017.2730359

1556-6013 © 2017 EU

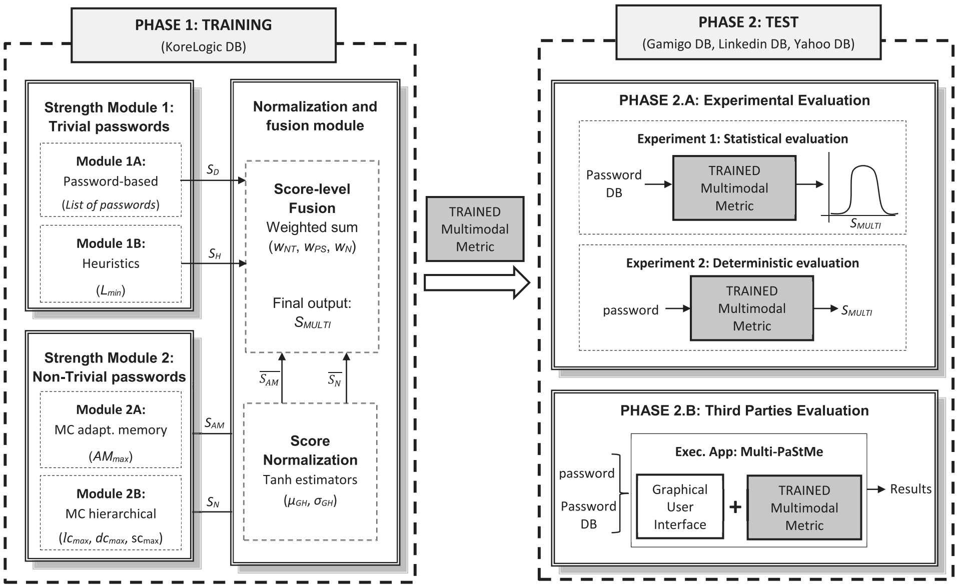

PHASE 1: TRAINING
(KoreLogic DB)
Strength Module 1: Trivial passwords
Module 1A: Password-based (List of passwords)
Module 1B: Heuristics (L_{min})
Strength Module 2: Non-Trivial passwords
Module 2A: MC adapt. memory (AM_{max})
Module 2B: MC hierarchical (lc_{max}, dc_{max}, sc_{max})
S_D
S_H
S_{AM}
S_N
Normalization and fusion module
Score-level Fusion
Weighted sum (w_{NT}, w_{PS}, w_N)
Final output: S_{MULTI}
$\overline{S_{AM}}$
$\overline{S_N}$
Score Normalization
Tanh estimators (μ_{GH}, σ_{GH})
TRAINED Multimodal Metric
PHASE 2: TEST
(Gamigo DB, Linkedin DB, Yahoo DB)
PHASE 2.A: Experimental Evaluation
Experiment 1: Statistical evaluation
Password DB
TRAINED Multimodal Metric
S_{MULTI}
Experiment 2: Deterministic evaluation
password
TRAINED Multimodal Metric
S_{MULTI}
PHASE 2.B: Third Parties Evaluation
Exec. App: Multi-PaStMe
password
Password DB
Graphical User Interface
+
TRAINED Multimodal Metric
Results

Fig. 1. Diagram of the general evaluation protocol followed to assess the multimodal strength meter introduced in [1].

been designed to produce objective comparable statistical results and may be regarded as a contribution to the field of password strength meters performance evaluation. 2) The new multimodal strength meter has shown its reliability as strength estimator over other existing state of the art algorithms. This has permitted to draw some useful conclusions regarding the selection of strong passwords and the design of password policies. 3) The novel multimodal algorithm has been integrated into an application that can be employed by users, providers and researchers to test the strength of passwords.[1]

The rest of the article is structured as follows. The general experimental protocol is presented in Sect. II. Then, each of the three phases comprised in the protocol (training, test and evaluation by third parties) are described in Sects. III, IV and V. Sect. VI contains the results of the two main experiments carried out in the test phase: experiment 1 in Sect. VI-A where the multimodal method is evaluated from a statistical point of view; and experiment 2 consisting of a deterministic analysis in Sect. VI-B. Sect. VII discusses how the new multimodal meter can drive users in real applications to select stronger passwords and how it can help to build new password policies. Finally, conclusions and a quick glance into the future are given in Sect. VIII.

II. Experimental Protocol

Following the argumentation given in the introduction, we have designed an experimental protocol to assess, from a general perspective, if the strength values assigned to passwords by the novel multimodal method follow a "sensible trend", that is, passwords that are easier to break are given on average lower scores and passwords difficult to break are given in general higher scores. As shown in Fig. 1, the protocol is divided in two main successive phases: training and test.

- **Phase 1: Training**. Used to fix the different parameters and transition matrices that define each of the modules that conform the complete multimodal method, as described in [1] (see the left panel in Fig. 1).

The test phase is, in turn, divided in two complementary initiatives, so that the specific version of the model trained in phase 1 is evaluated: A) based on two experiments described in the present article, and B) in the future through the test of third parties (see the right panel in Fig. 1).

- **Phase 2-A: Experimental evaluation**. Once the whole algorithm has been trained, it is validated following a general framework composed of two experiments: 1) Experiment 1 - Statistical (multimodal): analysis, from a probabilistic perspective, of the strength assigned by the proposed multimodal method to different sets of passwords, according to their resilience to typical guessing attacks; 2) Experiment 2 - Deterministic (multimodal): multimodal strength computation of some particular password examples.
  The performance of the new multimodal algorithm is also compared in both experiments to that of different meters from the state of the art.
- **Phase 2-B: Third parties evaluation**. As explained above, determining which of two strength meters is more accurate is a very difficult problem. For this reason, as a further evaluation strategy, the proposed multimodal strength estimator has been integrated into the closed application Multi-PaStMe (Multimodal - Password Strength Meter). This way, other researchers and developers can benefit from it in order to integrate it in their systems, or compare the results of the new method to their own password strength meters, in an on-going evaluation process.

Please be aware that, as will be explained below, the multimodal method contains algorithms fully related to the machine learning field. Therefore, should they be trained and tested using the *exact same* sets of passwords, the results would present a significant positive bias. To avoid such a situation,

[1] Please contact the authors for distribution details.

the two main phases, training and test, are carried out using independent password datasets (as indicated in Fig. 1). In order to ensure such independency, all test datasets contain passwords that were leaked *after* the publication of the training data (please see Sects. III and IV for further details).

However, using different datasets for training and test, does not mean that the *nature* of the data included in those datasets should be different. On the contrary, as in any other machine learning problem, if the training and test data are not of the same *type*, the final performance of the method will considerably worsen. As an easy example, if the multimodal strength meter is trained on English passwords and tested on Russian passwords written with the Cyrillic alphabet, the method will simply not work. This way, the same general context has to be defined both for the training and test data.

For the present article, the training and test phases have been carried out considering a typical and broad operational setting defined by: English-based application, not designed for a specific community but thought for users with a generic profile (e.g., online email), vulnerable to offline attacks, protected with a common hashing algorithm like SHA-3. All the training and test data used in the experiments comply, to a large extent, with this general setup.

The next sections describe the complete experimental protocol: 1) Training of the parameters that define the multimodal method (Sect. III); 2) Description of the different experiments that have been conducted to validate it (Sect. IV); 3) Implementation details of the Multi-PaStMe application for the on-going third parties evaluation process (Sect. V).

III. Phase 1: Training

As was highlighted in Part I [1] and also in the introduction of the present article, password strength is a highly application-dependent value. For instance, the strength of a given password can significantly vary depending on external factors such as: 1) the type of oracle disclosing the password (e.g., if offline attacks against hash values are permitted or only remote access with a limited number of guesses is possible); 2) the type of hashing or encryption algorithm used (e.g., attacking MD5 hashes is significantly faster than attacking SHA-3 ones); 3) the language (e.g., in general, a Hungarian word in a Spanish-based application will be a stronger password than that same password in a Hungarian-based system, and viceversa); 4) background of the users and/or context of the application where the password is being used (e.g., a password like "PioletIce" may be stronger for a webmail online application than for an online shop of mountain gear).

The multimodal method presented in Part I [1] can be adapted during the training phase to estimate the strength of passwords in very diverse contexts. To this end, the system is divided into three major modules, each of them containing two individual sub-modules (see the left panel in Fig. 1):

- Strength module 1: Trivial passwords. This module is designed to detect the two basic attacks that will be performed almost with all certainty at the beginning of any password guessing session: 1) attacks based on a list of the most used passwords and 2) brute-force attacks. To this end it contains, "Strength module 1A: Password-based" (which outputs the strength score S_D) and "Strength Module 1B: Heuristic-based" (which outputs the strength score S_H).
- Strength module 2: Non-Trivial passwords. This module is designed to cope with non-trivial passwords that are robust to attacks based on lists of common passwords or to brute-force attacks. Strength values are assigned according to the likelihood that a person would choose a given password. To this end, it includes two novel algorithms based on Markov Chains: 1) "Module 2A: Adaptive Memory Markov Chain" (which outputs the strength score S_{AM}) and 2) "Module 2B: Hierarchical Markov Chain" (which outputs the strength score S_N).
- Normalization and fusion module. The final objective of these two sub-modules is to combine the scores (S_D, S_H, S_{AM} and S_N) provided by the four individual strength estimation algorithms presented above, into the final unique multimodal score S_{MULTI}. To do this, techniques from the field of information fusion are used.

The process to adapt the multimodal meter to the specificities of a given application is accomplished by fixing (i.e., *training*) the parameters that define each of the four individual password strength modules, as well as the normalization and fusion modules. These parameters were summarized at the end of the sections dedicated to the description of each particular module in Part I. As such, we refer the interested reader to that Part I for a detailed explanation of the algorithms [1].

A. Training of Module 1: Trivial Passwords

This section describes the process followed for the selection of the parameters for the two modules used for the detection of trivial passwords within the multimodal meter.

1) Module 1A: Password-Based: Module 1A takes as input a password and gives as output a strength score S_D. This module is designed to detect attacks carried out using lists of popular passwords. The strength score S_D takes either value 0, for passwords present in a blacklist, or 10, for passwords resistant to the attacks (i.e., not present in the blacklist).

The input parameter that has to be defined for this module is a blacklist of passwords $List_{pwd}$. Three linked parameters were taken into account to take a decision on this list: 1) previous work has shown that a blacklist with as few as 1000 banned passwords is able to reduce the percentage of cracked passwords over 50 000 guesses from 25% to 20% [3]. 2) Large blacklists may be regarded as a big nuisance by end users [9]. 3) It should not be forgotten that the module is just one part of an overall multimodal strength meter. This way, it is thought to detect only the passwords that can be considered as "the worst of the worst". The final purpose is that such very reduced set of passwords can be rated as *trivial* or *very weak* by the global algorithm. The rest of passwords potentially present in a larger list will be detected by the other modules in the algorithm and rated as weak.

Given those premises, the blacklist considered for this module is formed by all different passwords that appear in: the list of 500 worst passwords published in 2008 [10], list of 370 passwords banned by twitter [11], and the list with the 100 most common passwords in the RockYou dataset [12].

Even considering the three previous requirements, the selection of one particular blacklist is a subjective decision. The reader should bear in mind that $List_{pwd}$ is just an input parameter of the model that can be modified according to the specific requirements of a given application (e.g., the blacklist for a French speaking on-line service will most likely differ from the one defined here, or a longer/shorter list could also be used).

2) Module 1B: Heuristic-Based: Module 1B takes as input a password and gives as output a strength score S_H. Its objective is to detect passwords that are vulnerable to brute-force attacks. The strength score S_H takes either value 0, for brute-forceable passwords, or 10, for passwords resistent to the attacks.

This module is totally application-dependent. In order to select the minimum length values of brute-forceable passwords for a given application, the key parameters that should be taken into account are (see [1] for further details): 1) amount of time that the theoretic brute-force attack will be running; 2) on-line or off-line attack; 3) salting; 4) hashing method; 5) number of characters contained in the alphabet from which passwords are selected.

Following previous works performing password attacks where a number of guesses between 10^{12} and 10^{14} was executed [13], [14], for this module we have considered as brute-forceable passwords those that may be broken in roughly 10^{13} guesses.

For a given alphabet with N symbols, there are N^L passwords of length L. This way, the minimum password length is defined by: $10^{13} = N^{L_{min}}$. Given this equation, the next values have been defined for the minimum length of passwords in module 1B:

- Passwords using all three character types (i.e., lower case, upper case, digits and special characters). The alphabet contains $N = 94$ characters: $L_{min} = 7$.
- Passwords formed by only lower case letters, only upper case letters, or only special characters. The alphabet contains (at least) $N = 26$ characters: $L_{min} = 9$.
- For only-digit passwords, the alphabet contains $N = 10$ characters: $L_{min} = 12$.
- For any pair-wise combination of the previous character classes the alphabet contains (at least) $N = 36$ characters: $L_{min} = 8$.

This means that passwords longer than L_{min} for each of the possible alphabets, are considered to be resistant to brute force guessing attacks.

B. Training of Module 2: Non-Trivial Passwords

The multimodal meter also integrates two submodules designed to estimate the strength of non-trivial passwords. These submodules are based on Markov Chains (i.e., algorithms related to machine learning) defined by transition probability matrices that require a lot of data to be reliably estimated. This way, both submodules have been trained using a publicly available dataset of 122 million English-based passwords released by KoreLogic in 2011 to support password related research. In the dataset, 83.5 million are unique passwords [15], [16]. This is, to the best of our knowledge, one of the largest password sets distributed to the password research community so far. For further details and statistics regarding the composition of the training dataset and of the structure of the passwords contained inside we refer the reader to Annex A, provided as accompanying material of the present article.

The dataset is the result of different data breaches that led to the release of hashes computed from passwords chosen by real users. Such data breaches are in general the result of vulnerabilities in the password storage system of a given company victim of a sophisticated hacking attack, and not of the strength of individual passwords selected by the users. However, once the password hashes are leaked, their robustness is put to test through off-line guessing attacks that are able to break the hashes coming from weak passwords that are then made available in plain text. This is also the origin of the three datasets used in the test phase (see IV). These datasets of password hashes are made available for research purposes with no link to any user information (e.g., real name, user name, email address).

As mentioned above, among other parameters, the KoreLogic password training set is essential to define the transition matrices of the Markov Chains integrated in the multimodal model.

1) Module 2A: Markov Chain With Adaptive Memory AM_m: This submodule takes as input a password and gives as output a strength score S_{AM}. This is a *local model* that searchers for specific word-related patterns within passwords.

The main parameter to be fixed for the model is the maximum memory size AM_{max} (see [1] for further details). There is not a unique optimal value that may be computed in a deterministic way for this parameter. Rather, its estimation should be done heuristically on a case by case basis. To do so it should be noticed that, essentially, the transition matrix T is equivalent to an exhaustive combination table in a search space defined by AM_{max} and N (i.e., all possible AM_{max} character combinations taken from a pool of N characters are reflected in the matrix). As the training data is limited, the larger AM_{max}: 1) the more zeros will populate the table; and 2) the fewer number of observations that will be used to compute the probability of non-zero occurrence sequences. In summary, for a finite and limited set of training data, the larger AM_{max}, the lower the statistical reliability of the very sparse matrix T.

Therefore, two linked factors should be taken into account in order to select AM_{max}: 1) Size of the training set: as a general rule, larger training sets will allow reliably training models defined by larger values of AM_{max}; 2) Size of T: the larger AM_{max}, the bigger the transition matrix T, eventually requiring a very large storing capacity and also slowing down the strength estimation process.

According to the trade-off that has to be reached between generality of the model, size of the transition probability matrix and accuracy, the maximum size of the memory, AM_{max}, is set to $AM_{max} = 4$. As mentioned above, this value is highly dependent on the size of the training KoreLogic dataset (i.e., 122 million passwords) and has been fixed by setting a sparsity threshold of 90% for the transition matrix, that is, no more than 90% of the transitions are null. A null transition is defined as any transition that has been observed 10 times or less.

In the case of a training set at least one order of magnitude larger, a model with memory five could be considered for the same sparsity threshold.

2) Module 2B: Hierarchical Markov Chain: It takes as input a password and gives as output a strength score S_N. This is a *global model* that accurately represents the general structure of human passwords.

The higher-order elements considered in the model belong to one of three classes: 1) letter-class: formed by letter subsequences of sizes 1 to lc_{max}; 2) digit-class: formed by digit subsequences of sizes 1 to dc_{max}; 3) special characters-class: formed by special characters subsequences of size 1 to sc_{max}.

The maximum lengths for each of the subsequences classes (lc_{max}, dc_{max}, and sc_{max}) were fixed according to their probability of occurrence in the KoreLogic dataset. Only subsequence lengths with a probability of occurrence higher than 0.1% were considered. Their final values are: $lc_{max} = 20$, $dc_{max} = 12$, and $sc_{max} = 6$. Therefore, the model is composed by a total $SS = 38$ higher-order subsequences.

The maximum number of subsequences that can form a password, P_{max}, so that it can be represented following the layered Markov Chain, was also determined according to the length distribution of the passwords in the KoreLogic dataset. It is set to $P_{max} = 15$, as passwords formed by a higher number of subsequences represent less than 0.01% of the total passwords in the training dataset (not enough data to reliably estimate their probability of occurrence).

C. Training of the Normalization and Fusion Module

The strength scores obtained from the previous four individual modules (i.e., S_D, S_H, S_{AM} and S_N) are very heterogeneous and should not be directly merged into a single multimodal value. Prior to their combination in the fusion sub-module, they need to be transformed into one common domain. This is accomplished through a process known as *score normalization*, which plays a very important role in the design of any *score level fusion* scheme.

The normalization submodule uses the *tanh estimators* in order to transform the scores from the different individual strength modules into the common range [0,10], prior to their fusion. The fusion sub-module, in turn, is based on the *weighted sum* to combine the normalized scores.

For the normalization submodule, the strength of the passwords in the KoreLogic dataset was computed according to the Markov Chain with Adaptive Memory and the Hierarchical Markov Chain. Those two sets of strength scores were used to determine the normalization parameters μ_{GH} and σ_{GH} for each of the two Markov-based algorithms.

On the other hand, the fusion weights to be used in the *weighted sum* were selected so as to: 1) give very low strength to trivial passwords; 2) have a balanced input from the two Markov-based models in the case of non-trivial passwords. This way, two different sets of weight values $[w_T, w_{AM}, w_N]$ are defined depending on the output of the two trivial password detectors (i.e., password-based and heuristic-based modules):

- $[w_T, w_{AM}, w_N] = [0.9, 0.05, 0.05]$ if $S_D = 0$ or $S_H = 0$.
- $[w_T, w_{AM}, w_N] = [0, 0.5, 0.5]$ if $S_D = 10$ and $S_H = 10$.

With these weight values, the strength of trivial passwords is restricted to the range [0,1], while non-trivial passwords can take any strength value in the range [0,10].

IV. Phase 2-A: Experimental Evaluation

The overall goal of the experimental evaluation phase is to assess if the multimodal meter trained in phase 1 is consistent in the assignment of strength values to passwords, that is, if it gives lower scores to passwords that are more easily cracked and higher scores to those that are harder to be broken (or that are subjectively regarded as stronger by humans).

With this objective in mind, two different experiments have been carried out:

- **Experiment 1: Statistical evaluation of the multimodal meter**. In this case, the goal of the experiment is to analyse, from a statistical perspective, the strength assigned by the multimodal method to sets of passwords with different levels of resistance to known attacks of increasing complexity. For reference, the strength estimation provided by the multimodal method is also compared to different meters from the state of the art: 1) the *de facto* password strength standard proposed by NIST [17]; 2) three different meters used by well-known large internet service providers such as Yahoo, Gmail and Dropbox. These last three meters are used as implemented in the publicly available PARS[2] application [18]. Further details about the implementation of this first statistical experiment are given below.
- **Experiment 2: Deterministic evaluation of the multimodal meter**. The statistical evaluation of the model carried out in experiment 1 follows a strict methodology and therefore may be understood as a consistent and general assessment. However, it fails to present factual results for individual passwords which can also be useful to illustrate the potential of the method. Following this reasoning, the goal of this experiment is to analyse the strength assigned by the proposed multimodal method to some specific password examples and to compare them to the same four meters from the state of the art considered in the previous experiment. This experiment should not be regarded in itself as a rigorous evaluation test (as it is based only on very few particular examples), but as a complement to the results presented in experiment 1.

A. Experiment 1: Test Datasets

The test passwords used in the statistical experiment (i.e., experiment 1) come from three data breaches that occurred after the publication of the KoreLogic dataset used for training. In particular, the password datasets used in the present work for testing have been regularly used in the literature for the development and analysis of studies related to password strength [18]–[20]:

[2] http://www2.ece.gatech.edu/cap/PARS/

- Gamigo dataset (data breach 2012): Gamigo is an online community of video-games players. It contains 7 million English-based plain-text passwords, 99.6% of which are unique. It shares 11.03% of its passwords with the training dataset.
- Linkedin dataset (data breach 2012): Linkedin is an online job-hunting platform. It contains 5.6 million English-based plain-text passwords, 90.8% of which are unique. It shares 14.85% of its passwords with the training dataset.
- Yahoo dataset (data breach 2014): Yahoo is a general online email service. It contains 740 000 English-based plain-text passwords, 77.3% of which are unique. It shares 26.56% with the training dataset.

For further details and statistics regarding the composition of the test datasets and of the structure of the passwords contained inside we refer the reader to Annex A, provided as accompanying material of the present article.

These three datasets have been chosen because, although all of them contain mainly English-based passwords, they come from applications that offer very different online services. This way, the profile of users, and therefore also the structure and strength of the passwords chosen, can be expected to differ to some extent. In particular, it is reasonable to assume that: Gamigo will contain the strongest passwords since gamers are in general aware of the best practices to select good passwords; Linkedin is a general platform where users introduce some potentially sensitive data, therefore, they may select, on average, stronger passwords to protect their accounts than in a very broad platform like Yahoo. This initial general assumption is supported by the statistics given in Annex A and also corroborated by the results obtained in the guessing session performed during the evaluation of the multimodal strength meter (see the next subsections and Table II).

It is important to highlight that, as specified above, the test datasets are partially included in the KoreLogic dataset used in the training phase (see Sect. III). In general, good practices in machine learning problems advise against this situation. The origin of such sensitive rule is that machine learning evaluation techniques try to reflect the reality of a given problem. In most cases, it is not rational or even possible that a sample used for training will later be processed by the algorithm during its regular operation. However, the case of passwords is a very particular one. It is a known fact that most available datasets share some passwords as a result of the human tendency to repeat its password selection [21]. Filtering out those common passwords would mean to artificially modify what happens in the real world, which is, in fact, what machine learning evaluation principles try to model. In that case, results would be negatively biased. Of course, on the other side of the spectrum, if the overlap between the training and the test data is too high, results would be equally unreliable. As such, a balance has to be achieved so that the percentage of common passwords shared by the training and test datasets reflects what could be expected in the real world. Although it is difficult to exactly determine such a balance, for the present work three different test datasets have been selected with a level of overlap that varies between 10% and 25%. An important characteristic to be taken into account is that all the data breaches that led to these three datasets were produced *after* the training dataset was released. Therefore, it is reasonable to assume that any possible overlap between them is the result of the natural human behaviour in password selection.

B. Experiment 1: Guessing Session

During experiment 1, the passwords contained in the three test datasets mentioned above have been grouped into different clusters depending on their resistance to attacks of increasing complexity. The clustering process has been designed to mimic a typical password guessing session such as the ones described in [22] and [23], where an attacker has off-line access to hash values corresponding to passwords. In such guessing sessions, the attacker attempts to sequentially retrieve the passwords starting by the most straightforward attacks and gradually moving towards the more complex ones. This means that the first attacks to be carried out are those that have traditionally shown a higher success rate measured in terms of passwords cracked per given number of attempts. These initial attacks are very fast retrieving passwords at the beginning but also become unsuccessful soon. The last attacks to be implemented are the most general ones, that is, those that are capable of cracking new passwords that were resistant to the previous cracking techniques but that, in turn, also generate many more wrong guesses which makes them less efficient. Such a password guessing strategy has even been implemented in automatic tools that sequentially launch the most efficient attack as the number of guesses increases [24], [25].

In particular, the guessing session performed for the present work is composed of three sequential steps, each of them comprising a category of attacks more complex than the previous step. This process divides each test dataset into four clusters of passwords, one for each set of passwords broken in every step of the guessing session and the final one containing the passwords that have not been retrieved by any of the attacks. The clusters have no overlapping as attacks have been applied in a successive manner, that is, each attack is launched only against the passwords not recovered by the previous attack/s.

Once each of the three test datasets has been divided into the four password clusters, the strength of each password is computed in order to analyse if, from a statistical point of view the strength assigned by different meters reflects the complexity of the attack that recovered them. The rationale behind such assessment approach is that a good password strength meter should correctly reflect the resistance of a password against actual password guessing attacks.

As a graphical aid, a general diagram of the clustering protocol followed is given in Fig. 2, where it can be seen that the three steps that compose the guessing session correspond to a specific category of known common attacks: 1) Step 1: Contains attacks that perform basic exhaustive searches using different combinations of alphabets and lengths; 2) Step 2: Consists of standard dictionary attacks without rules; 3) Step 3: Encloses more complex dictionary attacks using word mangling rules.

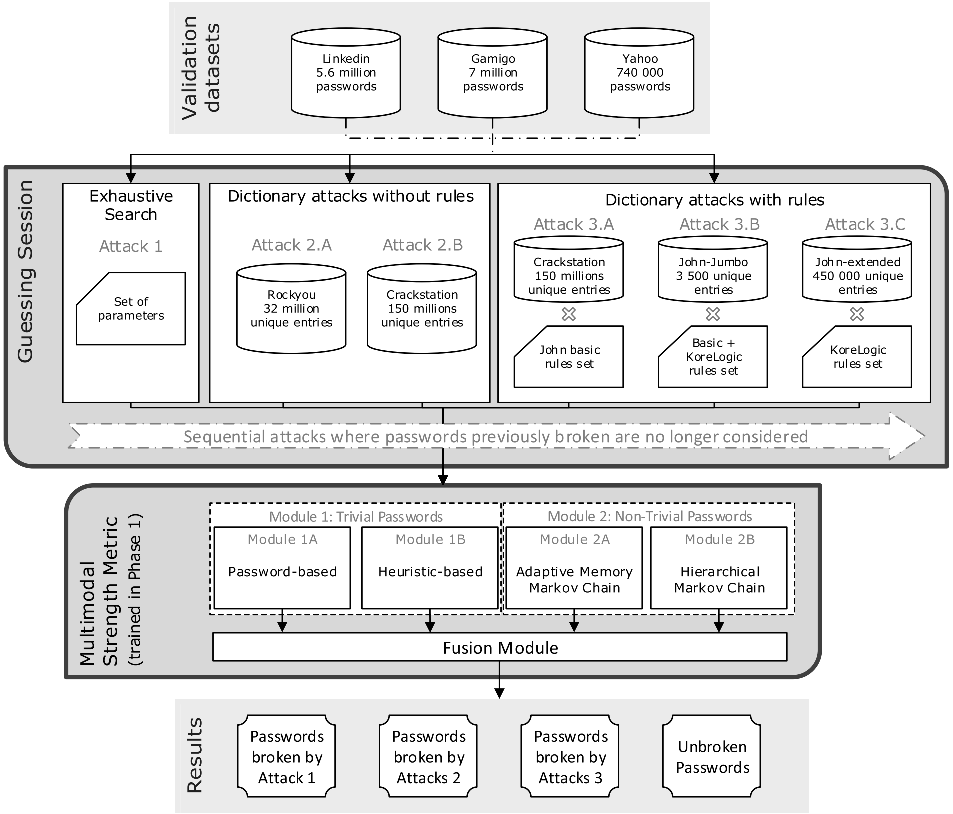

Validation datasets
Linkedin 5.6 million passwords
Gamigo 7 million passwords
Yahoo 740 000 passwords
Guessing Session
Exhaustive Search
Attack 1
Set of parameters
Dictionary attacks without rules
Attack 2.A
Attack 2.B
Rockyou 32 million unique entries
Crackstation 150 millions unique entries
Dictionary attacks with rules
Attack 3.A
Attack 3.B
Attack 3.C
Crackstation 150 millions unique entries
John-Jumbo 3 500 unique entries
John-extended 450 000 unique entries
John basic rules set
Basic + KoreLogic rules set
KoreLogic rules set
Sequential attacks where passwords previously broken are no longer considered
Multimodal Strength Metric (trained in Phase 1)
Module 1: Trivial Passwords
Module 2: Non-Trivial Passwords
Module 1A Password-based
Module 1B Heuristic-based
Module 2A Adaptive Memory Markov Chain
Module 2B Hierarchical Markov Chain
Fusion Module
Results
Passwords broken by Attack 1
Passwords broken by Attacks 2
Passwords broken by Attacks 3
Unbroken Passwords

Fig. 2. Diagram of the process followed in Experiment 1 of the test phase, for the statistical assessment of the multimodal strength meter.

Please note that the objective of the experiment is to simulate a plausible password guessing session with some realistic typical attacks, in order to analyse the strength of the passwords broken at each step. We are aware that other types of attacks not considered in this experiment could have been included. However, we do feel that the set of attacks executed can be seen as a good baseline example of an illustrative password guessing session, and that they can therefore serve their purpose: determine if the proposed multimodal meter assigns meaningful strength values to passwords.

In the following subsections we give the specific implementation details of the attacks carried out in the three steps that conform the password guessing session. Prior to that, we present here some general characteristics common to the whole guessing session:

- Number of guesses (NoG). Following previous works simulating guessing sessions [13], [14], where a number of guesses between 10^{12} and 10^{14} was performed, a limit in the range of 10^{13} guesses was set for the attacks. This means that, for each of the three steps in the guessing session (i.e., brute-force attacks, dictionary attacks and dictionary with rules attacks), a configuration of the attacks is chosen so that the total sum of all guesses performed in that step falls within the vicinity of $NoG_T = 10^{13}$ guesses.
- Hardware: The password guessing session was performed on a platform of common hardware: Intel Xeon E5-2670 PC running under Ubuntu 14.04 with two AMD Radeon R9 290 cards. Under this specific implementation, considering SHA-3 hashing and an off-line attack, it takes roughly one day to perform the estimated limit of $NoG_T = 10^{13}$ guesses.

  Please consider that this attacking time (one day) is only a theoretical laboratory exercise for the purpose of the present work. In a practical system, this time can be largely increased by a service provider should he follow good password practices like using a slower hashing function or monitoring data breaches and forcing password resets if a breach is detected. In that case, a much lower number of guesses for the attacks would be feasible (resulting in much fewer cracked passwords).
- Software: All the attacks have been carried out using the John-the-Ripper open source software [25].

1) Step 1: Exhaustive Search Attacks: The first step of the sequential guessing process is composed of a series of exhaustive search attacks, also known as brute-force attacks, using the alphabets and password lengths specified in Table I. Table I should be interpreted as follows: taking for instance the first row, the corresponding attack generates as number of guesses NoG *all* possible combinations up to length $L = 11$ that can be produced with the alphabet of $N = 10$ digits, that is, $NoG = N^L$. An analogue interpretation is valid for the remaining rows.

The length values for each of the alphabets are selected in order to comply with the approximate limit of $NoG_T = 10^{13}$ guesses fixed for each step of the attacks (see introduction

TABLE I

ALPHABETS AND MAXIMUM PASSWORD LENGTH CONSIDERED IN THE SET OF BRUTE-FORCE ATTACKS CARRIED OUT AS PART OF THE FIRST STEP OF THE PASSWORD GUESSING SESSION THAT WAS CONDUCTED DURING THE STATISTICAL EVALUATION OF THE MULTIMODAL STRENGTH METER. THE LAST COLUMN INDICATES THE NUMBER OF GUESSES (NoG) CARRIED OUT IN EACH OF THE ATTACKS

Attack	Alphabet	Max Length	NoG
1.A	10 digits	11	10^{11}
1.B	26 lower case	8	2.1×10^{11}
1.C	26 upper case	8	2.1×10^{11}
1.D	32 special	8	1.1×10^{12}
1.E	2-combination previous 4	7	5.8×10^{12}
1.F	All 96 ASCII characters	6	7.8×10^{11}

to Sect. IV). Adding up the last column in Table I, the total number of guesses performed in this step 1 of the guessing session is 8.2×10^{12}.

It is important to note that, since the lengths shown in Table I for brute-forceable passwords coincide with the parameters selected in the training phase for module 1B (i.e., heuristic-based), the multimodal meter will be highly efficient in the detection of passwords vulnerable to exhaustive search attacks. If this was not the case, some trivial passwords would be given a higher strength than expected or, the other way around, some strong passwords a lower score. Therefore, in order to have a good detection rate of passwords that can be guessed using these attacks, it is very important to select the appropriate minimum length parameters in module 1B for each particular application (i.e., taking into account parameters such as: time considered for a brute-force attack, hashing method, salting, on-line/off-line attacks, etc).

2) Step 2: Dictionary Attacks Without Rules: The second step of the guessing session is based on the use of plain wordlists. This process is also known as *dictionary attack*. Two wordlists are used for this step, each of them containing a larger number of guesses.

- Attack 2.A: Rockyou wordlist (basic wordlist). It is a very well known dataset used in many password related works containing 32 million unique passwords.
- Attack 2.B: Crackstation wordlist (extended wordlist). This wordlist contains passwords from different datasets (none of the three test datasets used) as well as every word from wikipedia and several well known books from the Gutenberg Project. It is composed of 150 million words (i.e., candidate passwords) [26].

This way, the total number of guesses generated in this step 2 of the guessing session is 1.82×10^8 (total number of words contained in both wordlists).

3) Step 3: Dictionary Attacks With Rules: In this step, a dictionary attack is again applied using several wordlists but this time combined with word mangling rules that modify each candidate password in a predefined way (e.g. append a number or a date at the end, capitalizing letters, etc.) Each rule increases the number of candidates to be evaluated, sometimes drastically, therefore increasing the overall complexity and length (i.e., number of guesses) of the attack. To keep the process within the approximate limit of guesses set for the different steps in the guessing session, $NoG_T = 10^{13}$, it was necessary to adapt the size of the wordlist to the complexity of the rule set (i.e., more complex rule sets are combined with smaller wordlists).

TABLE II

PERCENTAGE OF BROKEN PASSWORDS FOR THE THREE EVALUATION DATASETS IN EACH OF THE THREE STEPS (I.E., S1, S2 AND S3) OF THE GUESSING SESSION DESCRIBED IN SECT. IV. THE LAST COLUMN SHOWS THE PERCENTAGE OF NON-BROKEN PASSWORDS

	TEST DATASETS - Broken passwords			
	Broken S1	Broken S2	Broken S3	Non-broken
Gamigo	16.5%	8.5%	4.7%	70.4%
Linkedin	24.9%	11.4%	7.6%	56.1%
Yahoo	42.5%	21.2%	8.6%	22.5%

- Attack 3.A: Extended wordlist with basic set of rules. This attack uses the basic set of rules provided with the tool John-the-Ripper combined with the Crackstation wordlist. This leads to a total 7.9×10^{12} guesses.
- Attack 3.B: Reduced wordlist with full set of rules. This attack uses the full set of rules defined in the Jumbo version which contains a community extended version of the John-the-Ripper basic set of rules. The wordlist used in this step is the one included in the John-the-Ripper tool that contains 3,500 passwords. This leads to a total 2.1×10^{13} guesses.
- Attack 3.C: Medium-size wordlist with extended set of rules. This attack uses the advanced set of rules of KoreLogic [27] with a wordlist significantly larger than the very reduced one used in attack 3.B. In order to perform this attack we used the wordlist provided by the John-the-Ripper community composed of 450 000 unique passwords. This leads to a total 1.1×10^{12} guesses.

Therefore, the total number of guesses performed in step 3 of the guessing session is 3×10^{13}.

As explained above, the three steps were conducted in a sequential manner, that is, each attack was only carried out on the passwords from the test datasets *not* retrieved by the previous attacks. This way, the test sets are divided into three subsets: passwords broken in step 1, passwords broken in step 2 and passwords broken in step 3. Finally, the fourth set of passwords comprises those passwords that were not retrieved during any of the three steps of the guessing session described above. The percentage of passwords contained in each of the four clusters for the three test datasets is given in Table II. These results confirm the hypothesis made at the beginning that the three databases clearly contain passwords with different levels of strength: Gamigo contains strong passwords, Linkedin average ones and Yahoo weak ones.

V. PHASE 2-B: THIRD PARTIES EVALUATION

The set of tests described in the experimental evaluation (phase 2-A, see Sect. IV) provide a realistic snapshot of the capabilities of the proposed multimodal meter. However, it is true that: 1) other test datasets and/or attacks could have been carried out in experiment 1; and 2) other particular password examples could have been chosen in experiment 2.

The reader will understand that it is not feasible to cover here all specific possibilities for those two experiments. Rather, phase 2-A should be seen as an example of a general evaluation framework that can be used to assess the performance of strength meters.

In this regard, in order to facilitate the research community the possibility to perform their own specific experiments, the proposed multimodal password strength method has been implemented in the executable application: Multimodal - Password Strength Meter (Multi-PaStMe). Multi-PaStMe is made available free-of-charge as a closed executable available upon request. The application accepts as input: 1) Individual passwords introduced through its graphical interface; in this case their strength value is computed in real time and displayed on the screen. 2) A list of passwords in a plain .txt file whose path is provided through the graphical interface; in this case the output is another plain .txt file with a two-column format: password *tab* strength.

There are three main objectives for Multi-PaStMe, all of them related to the creation of a third parties on-going evaluation process of the multimodal meter that can complement the results already obtained in the experiments of phase 2-A (see Sect.IV). The three goals may be summarized as follows: 1) Provide the interested reader with an easy tool to personally assess the method by analyzing to what extent it produces sensible strength estimations for specific password examples; 2) Provide researchers with a strength meter that can be used as a baseline result with which to compare future developments in the field (both from a deterministic and statistical perspective); 3) Provide application developers with an easily integrable tool that can give useful real-time feedback to users regarding the strength of their passwords.

As an estimation of its speed to be used in real applications, the current version of Multi-PaStMe takes on average 0.02 seconds to evaluate the strength of a password. This execution time has been achieved on a standard Intel Xeon E5-2670 PC running under Ubuntu 14.04.

As already mentioned in Sect. III, it is important to notice that the multimodal method evaluated in this article is general in the sense that it can be explicitly adjusted to better measure the password strength in different application-specific environments. This adaptation process is carried out at the training phase and depends on: 1) the values assigned to the different parameters of each of the modules and 2) the dataset used to train the transition matrices of the different Markov-based models.

The application Multi-PaStMe contains the pre-trained version of the method described in Sect. III. Therefore, it has been adjusted to estimate the strength of passwords in a quite generic application defined by the parameters: English-based, not designed for a specific community but thought for general users (e.g., online email), vulnerable to offline attacks, relatively simple hashing algorithm such as SHA-3.

In the case that future research works would like to present comparative results with respect to the proposed multimodal meter, the authors may use the current version of Multi-PaStMe to compute the multimodal strength values on their own test datasets. However, please be aware that, if the same dataset that was utilized for training (i.e., KoreLogic), is considered during the assessment process, results may be biased. Therefore, it is recommended that other password datasets different from KoreLogic are employed for testing. It is also recommended that the test datasets come from data breaches that occurred after 2011 (year in which KoreLogic was released) in order to avoid excessive overlap between the training and test data (as explained in Sect. IV).

As part of future work, a trainable version of the application will be generated so that, if required, it can be adapted to the necessities of each user.

VI. Results

This section presents and analyses the results of the two experiments carried out in the test phase described in Sect. IV.

A. Experiment 1: Statistical Evaluation

As a first evaluation experiment, the multimodal strength meter was computed for each of the four password sets in which the three test datasets (i.e., Gamigo, Linkedin and Yahoo) were divided following the 1-day password guessing session described in Sect. IV.

In order to have a comparison between the new multimodal approach and other existing state of the art meters, the password strength of the test datasets was also computed according to: 1) the *de facto* standard specified by NIST [17], which has been normalized to the range [0,10] using the same procedure as the multimodal score to help the comparison between the two (see the section describing the fusion module in Part I [1] for further details on the normalization algorithm); 2) three password-checkers used by well-known large internet service providers such as Yahoo, Gmail and Dropbox [28], that have been considered in previous works comparing the efficiency of different existing password strength estimators [2], [4], [5]. These last three operational meters have been computed according to their implementation in the free available tool PARS, which integrates several individual strength estimation algorithms [18]. While it is true that the experimental comparison could have been extended to other Markov-based strength estimation methods previously proposed [4], [20], it was not possible to find any public and working implementation of those algorithms, making the experimental comparative task highly difficult. However, a comparison from a theoretical standpoint can be found in Part I [1]. For further discussion, we refer the interested reader to Annex B, provided as accompanying material of the present paper, where additional evaluation experiments of the two individual Markov-based modules, 2A and 2B, are described. In any case, we believe that the state of the art meters selected for the present experimental evaluation are good illustrative examples that can show the strengths and limitations of the proposed multimodal meter.

The multimodal strength distributions of the password sets is plotted on the first row of Fig. 3, whereas the strength distributions corresponding to NIST, Yahoo, Gmail and Dropbox appear on the remaining four rows. Each column in the figure shows the strength distributions for the three test datasets. Please be aware that, while the multimodal meter and NIST's meter give as output real numbers between 0 and 10 (once normalized), the three commercial estimators produce only a fixed number of discrete strength levels (e.g., for the Dropbox meter these levels are: 'very weak', 'weak', 'so-so', 'good' and 'strong'). For this reason, the multimodal meter and NIST's meter have been plotted as continuous distributions, while the three commercial meters appear with bar-plots.

In principle, according to the level of difficulty to retrieve the passwords, a good strength meter should assign on average: 1) the lowest score to passwords broken in the first step of the guessing session (comprising the simplest brute-force attacks), plotted in black in Fig. 3; 2) a higher score to passwords broken during the second step (dictionary attacks without rules), plotted with dark-grey in Fig. 3; 3) still a higher score to passwords retrieved in the third step (consisting of the most advanced attacks performed, dictionaries with rules), plotted with light-grey in Fig. 3; 4) finally, the highest average strength value should be given to non-broken passwords, plotted with the lightest shade of grey in Fig. 3.

Therefore, in Fig. 3, a lighter shade of grey means that the passwords were broken in a later step of the guessing session, or, in other words, a lighter grey corresponds to theoretically stronger passwords. As such, a consistent strength meter should assign higher values to lighter-grey distributions. As such, the way to interpret Fig. 3 is that a good meter should comply with the general principle: "left-dark and right-light".

The strength distributions depicted in Fig. 3 can already give a general idea of the correspondence between the robustness of passwords to attacks of increasing complexity and the strength assigned by a given meter. However, as a way to complement these plots and in order to present a more objective assessment than the mere visual comparison, the level of overlap between the strength distributions is computed according to the *Kullback-Leibler divergence*, $KL(P\|Q)$. This is a distance-like meter that gives an estimation of the dissimilarity between two statistical distributions P and Q [29]. The more separated the distributions are, the higher the K-L value is, ranging from 0, when the two distributions fully coincide, to infinite, when there is no overlap between them.

Table III contains the K-L divergence between: A) the distribution of non-broken passwords (i.e., distribution in the lightest grey in Fig. 3), with respect to each of the three distributions corresponding to passwords broken in the guessing session using B1) brute-force attacks (KL_{BF}), B2) dictionary attacks (KL_D) and B3) dictionary with rules attacks (KL_{DR}). The K-L divergence for the three commercial meters has been obtained by assigning a numerical value in a linear scale from 0 to 10 to each of the discrete strength levels. For example, in the case of the Dropbox meter, the numerical values are: 'very weak'=0, 'weak'=2.5, 'so-so'=5, 'good'=7.5 and 'strong'=10.

TABLE III

RESULTS FROM EXPERIMENT 1 (SEE SECT. VI-A). KULLBACK-LEIBLER DIVERGENCE BETWEEN THE STRENGTH DISTRIBUTIONS OF NON-BROKEN PASSWORDS AND PASSWORDS BROKEN BY BRUTE-FORCE ATTACKS (KL_{BF}), DICTIONARY ATTACKS (KL_D) AND DICTIONARY+RULES ATTACKS (KL_{DR}). THEREFORE, THESE VALUES CORRESPOND TO THE DISSIMILARITY BETWEEN THE STRENGTH DISTRIBUTION PLOTTED IN THE LIGHTEST GREY IN FIG. 3 (I.E., DISTRIBUTION CORRESPONDING TO THE NON-BROKEN PASSWORDS) AND THE OTHER THREE DISTRIBUTIONS. THE LAST COLUMN SHOWS THE OVERALL GOODNESS SCORE (OGS) OBTAINED BY THE FIVE CONSIDERED METERS

	MULTIMODAL - KL Divergence			
	B-F (KL_{BF})	Dict (KL_D)	Dict+R (KL_{DR})	OGS
Gamigo	6.23	2.05	1.56	**19.92**
Linkedin	5.84	1.54	0.44	**3.95**
Yahoo	5.73	0.70	0.50	**2.00**

	NIST - KL Divergence			
	B-F (KL_{BF})	Dict (KL_D)	Dict+R (KL_{DR})	OGS
Gamigo	1.50	0.50	0.08	**0.06**
Linkedin	1.82	0.38	0.16	**0.11**
Yahoo	3.26	0.15	0.05	**0.02**

	YAHOO - KL Divergence			
	B-F (KL_{BF})	Dict (KL_D)	Dict+R (KL_{DR})	OGS
Gamigo	0.96	0.17	0.02	**0.003**
Linkedin	0.84	0.25	0.26	**-0.057**
Yahoo	0.41	0.14	0.11	**0.006**

	GMAIL - KL Divergence			
	B-F (KL_{BF})	Dict (KL_D)	Dict+R (KL_{DR})	OGS
Gamigo	1.42	0.70	0.02	**0.012**
Linkedin	1.49	0.69	0.007	**0.007**
Yahoo	2.08	0.57	0.02	**0.020**

	DROPBOX - KL Divergence			
	B-F (KL_{BF})	Dict (KL_D)	Dict+R (KL_{DR})	OGS
Gamigo	2.29	1.84	0.81	**3.41**
Linkedin	1.22	0.91	0.12	**0.13**
Yahoo	1.41	0.48	0.21	**0.14**

A good strength meter should comply with two conditions regarding the three values of the Kullback Leibler divergence presented above:

- Condition 1: As a general rule, the higher the K-L divergence value between any of the broken passwords distributions and the non-broken passwords distribution, the better the corresponding strength meter. That is, a good strength meter should present as high values as possible for KL_{BF}, KL_D and KL_{DR}.
- Condition 2: A good strength meter is expected to present less overlap between the distribution of non-broken passwords and the distribution of passwords broken with the simplest attack (i.e., brute-force), than with the distribution of passwords retrieved by a complex attack (i.e., dictionary with rules). Therefore, in Table III, a decreasing value of the K-L divergence should be observed from column 2 (passwords broken with the brute-force attack) to column 3 (passwords broken using the dictionary with rules attack). That is, a good strength meter should comply with: $KL_{BF} > KL_D > KL_{DR}$.

As a way to reflect the previous two conditions in just one objective quantitative value, a new Overall Goodness Score (OGS) is introduced as an assessment tool to easily compare different strength meters. This overall score is computed as: $OGS = KL_{BF} \times KL_D \times KL_{DR}$. It is positive if condition 2 holds and negative otherwise. The larger the value of OGS, the better the strength meter. The rationale behind the use of the product of all three individual K-L divergences, and not their sum, is that, this way, to obtain a high overall

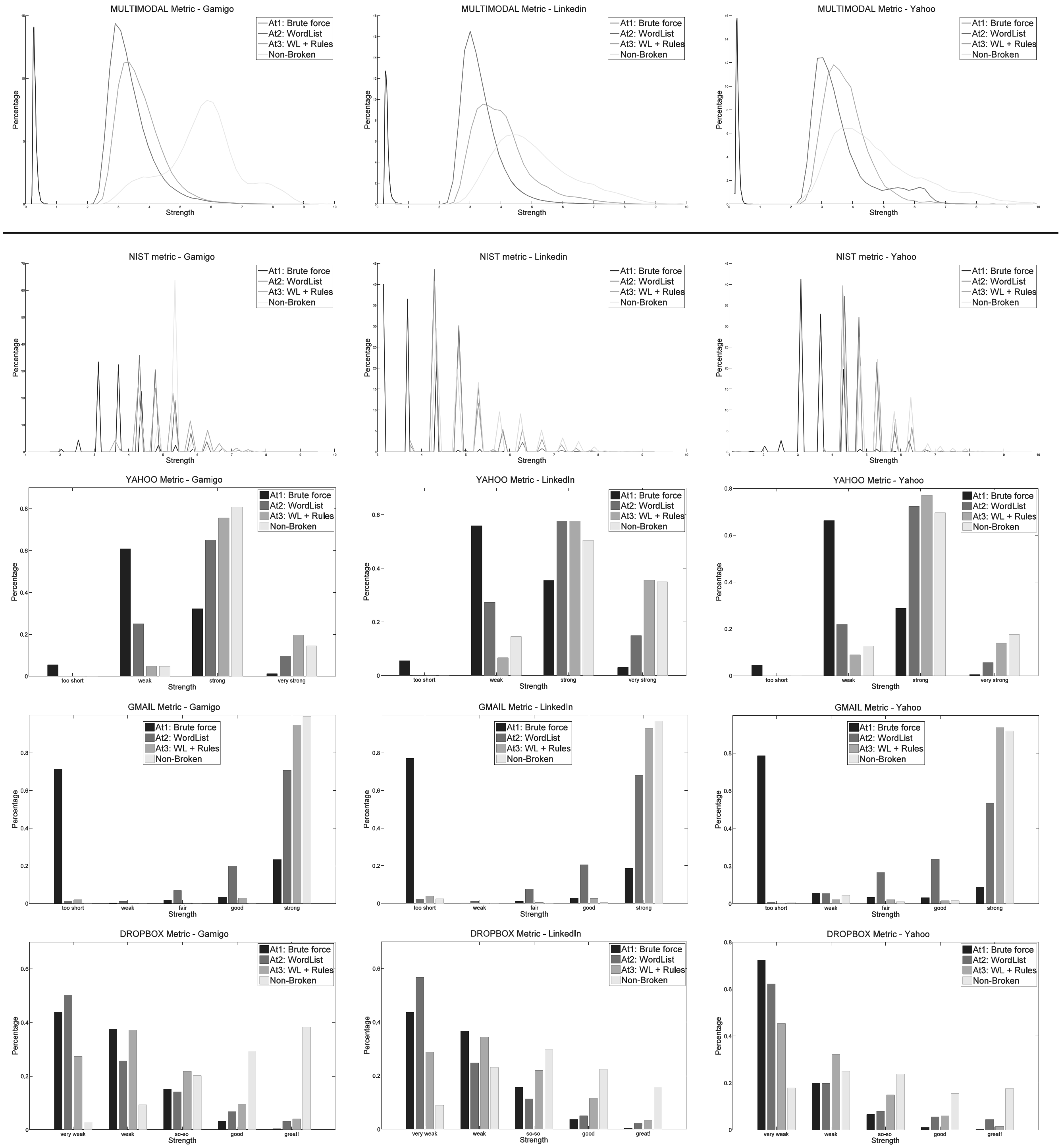


Fig. 3. Results from experiment 1 (see Sect. VI-A). Strength distributions corresponding to the four password clusters in which the test datasets were divided according to the robustness of passwords to attacks of increasing complexity (see Sect. IV for further details). The password strength distributions in the: 1) first row were computed according to the proposed multimodal meter; 2) second row correspond to the NIST recommendation; 3) third row were computed using the meter from Yahoo; 4) fourth row according to the meter from Gmail; 5) fifth row according to the Dropbox meter. The three columns correspond to a different test dataset: 1) first column corresponds to the Gamigo dataset; 2) second column to the Linkedin dataset; and 3) the third column to the Yahoo dataset.

score, it is not enough to get one good individual score (e.g., KL_{BF}) and two very low ones (e.g., KL_D and KL_{DR}), as this would still lead to a low final score. This would be the case, for instance, of a meter that is very good at detecting brute-forceable passwords and bad at detecting passwords vulnerable to dictionary-based attacks. Using the product to compute OGS, forces the meter to show a good *global* performance at detecting weak passwords, independently of the attack.

The OGS is an attempt to summarize the *global* behavior of a strength meter in just one value. However, as any summarization exercise, it can miss certain strengths or flaws of the method, such as: if it is good at detecting a specific type of vulnerable passwords or if it presents a tendency to overestimate or underestimate the strength of passwords. Therefore, the OGS should be understood as a quick and easy comparative tool among meters, able to give a fair estimation of which method has a better overall performance. However, for a complete analysis of a given meter, the results that appear in both Fig. 3 and Table III should also be presented.

Keeping in mind the general patterns for well-behaved strength meters described above both for the distributions and for the Kullback-Leibler divergence, we present in the following an analysis of the results of the five different methods considered in the experiment:

1) Multimodal Meter: Shown in row 1 of Fig. 3. The proposed meter follows exactly the expected behaviour described above for good strength estimators, that is, the four distributions are gradually plotted from left (lowest score, darkest grey shade, weakest passwords) to right (higher score, lighter grey shade, stronger passwords). This result shows that, the password robustness to attacks of increasing complexity and the actual strength value assigned by the novel multimodal meter are highly correlated for all three test datasets.

This result is confirmed by the K-L divergence values shown in Table III, where it may be seen that for all three test datasets, the condition $KL_{BF} > KL_D > KL_{DR}$ holds, while maintaining the highest individual values of all five meters for the three test datasets. This translates into the highest OGS.

As already mentioned in Sect. IV, it should be noticed that the parameters selected for module 1B (i.e., designed to identify passwords vulnerable to brute-force attacks) are exactly those considered for the brute-force attacks carried out in the evaluation guessing session. Therefore, it could be rightly argued that the very good results of the multimodal meter in the detection of brute-forceable passwords (see KL_{BF} in Table III) are positively biased. While this is true, it should also be noticed that the multimodal meter obtains as well the best strength estimation of passwords broken by dictionary and dictionary with rules attacks, in all the evaluation datasets (see KL_D and KL_{DR}). This way, even if the detection of brute-forceable passwords worsened, it is safe to assume that the multimodal meter would still present the best overall behaviour of all the tested meters.

2) Nist Meter: Shown in row 2 of Fig. 3. Following NIST's recommendation [17] (currently used in many practical systems), only certain real strength values are possible, which derives in "spiky" strength distributions difficult to interpret. However, the K-L divergence values presented in Table III show that, in spite of many justified criticisms [3], [30], the NIST meter has, from a statistical perspective, a quite reasonable behaviour: in all cases the K-L divergence decreases from the easiest passwords to break (those vulnerable to brute force attacks), to passwords retrieved by more complex attacks (dictionary with rules).

From this statistical analysis, the main limitations that can be pointed out are that the meter tends to: 1) on the one hand, overestimate the strength of short weak passwords; 2) on the other hand, underestimate the strength of robust passwords: the strength difference between passwords guessed by dictionary-based attacks and non-broken passwords is almost negligible. These observations reinforce the conclusions of previous evaluations of the same meter [3].

Even if, from an overall statistical perspective, the NIST meter is quite consistent, it still fails to produce logical estimations for certain types of passwords, as will be shown in experiment 2.

3) Yahoo Meter: Shown in row 3 of Fig. 3. This meter considers the strength levels: 'too short', 'weak', 'strong' and 'very strong'. It can be seen that the expected tendency "left-dark and right-light" is not really clear. In fact, for the linkedin DB it presents a negative OGS value, which means that the expected condition $KL_{BF} > KL_D > KL_{DR}$ does not hold as $KL_D < KL_{DR}$.

Its biggest limitation is that it presents a marked inclination to overestimate the strength of passwords vulnerable to dictionary-based attacks, which are marked in a majority as 'strong' (third out of four strength levels).

4) Gmail Meter: Shown in row 4 of Fig. 3. This meter considers the strength levels: 'too short', 'weak', 'fair', 'good' and 'strong'. As can be seen looking at the KL_{DR} column in Table III, this estimator has still a larger inclination than the yahoo meter to overestimate the strength of passwords vulnerable to dictionary-based attacks (it presents significantly lower KL_{DR} values).

As can be seen in Table III, this meter is able to correctly produce a very marked strength gap between brute-forceable passwords and the rest. However, the strength difference between passwords vulnerable to dictionary attacks, to dictionary attacks with rules and non-broken passwords is hardly noticeable, being all assigned a very high robustness (i.e., very large overlap between distributions reflected in the values of KL_D and KL_{DR}).

5) Dropbox Meter: Shown in row 5 of Fig. 3 and in the last section of Table III. This meter considers the strength levels: 'very weak', 'weak', 'so-so', 'good' and 'great!'. From all four state of the art meters, this is the most consistent one, presenting a good correlation between the resistance of passwords to attacks and their assigned strength.

As has been already highlighted, one of the advantages of the new overall score OGS is that it allows comparing in a fast and quantitative manner different strength meters. Based on this score, the ranking of the five considered strength estimators would be leaded by the multimodal meter (highest OGS for all three evaluation datasets), followed by (in this order): Dropbox, NIST, Gmail and Yahoo.

The new OGS should be regarded as a convenient, quick and fairly reliable estimator of the *global* performance of strength meters. However, it may fail to detect a specific behaviour of a given method. For instance, a certain algorithm A with a lower OGS than a different algorithm B may be, however, more efficient at detecting a specific set of weak passwords (e.g., brute-forceable). As such, it is advised to accompany any ranking based on the OGS with complementary results like those shown in Fig. 3 and Table III.

As mentioned in the introduction of Sect. II, determining which of two password strength estimators presents a better performance is not an easy task. Except for very clear cases

TABLE IV

Results from Experiment 2 (See Sect. VI-B). Some Individual Password Examples With Their Strength Score Computed With: 1) the Multimodal Meter; 2) NIST Meter; 3) Yahoo Meter; 4) Gmail Meter; 5) Dropbox Meter. For the Three Commercial Checkers, in Parenthesis Appears the Rank of the Password Out of the Total Number of Strength Levels Considered by That Particular Meter. ID is Just an Identification Number Given to Each Password for Quick Reference in Sect. VI-B Where Results are Analysed

ID	Password example	**MULTI**	**NIST**	**YAHOO**	**GMAIL**	**DROPBOX**
1	password	0.25	4.37	Weak (2/4)	Weak (2/5)	Very weak (1/5)
2	mypassword	2.67	5.37	Weak (2/4)	Weak (2/5)	Very weak (1/5)
3	mypasswordrocks	4.79	7.59	Weak (2/4)	Strong (5/5)	Very weak (1/5)
4	mypasswordrock2	5.16	7.59	Strong (3/4)	Strong (5/5)	Very weak (1/5)
5	myp4sswordrocks	5.65	7.59	Strong (3/4)	Strong (5/5)	Very weak (1/5)
6	mypasswordrock$	5.35	7.59	Strong (3/4)	Strong (5/5)	Very weak (1/5)
7	mypasswordrockS	5.66	7.59	Strong (3/4)	Strong (5/5)	Very weak (1/5)
8	MyPasswordRocks	6.36	7.59	Strong (3/4)	Strong (5/5)	Very weak (1/5)
9	MyPassw0rdR0cks	7.06	7.59	Very strong (4/4)	Strong (5/5)	Very weak (1/5)
10	MyPassw0rdR0ck$	7.07	8.75	Very strong (4/4)	Strong (5/5)	Very weak (1/5)
11	mypasswordrocksreallyalot	9.09	9.43	Weak (2/4)	Strong (5/5)	So-so (3/5)
12	y94r6f9k	7.10	4.37	Strong (3/4)	Strong (5/5)	Great! (5/5)
13	y94R(f%K	9.10	6.34	Very strong (4/4)	Strong (5/5)	Great! (5/5)

(e.g., Gmail or Yahoo), it is difficult to give a clear categorical answer. However, the experimental protocol followed in this experiment 1, together with the method to report results based on Fig. 3, its accompanying Table III and the overall goodness score OGS, can be regarded as a good example of a general standard methodology to perform such a comparison. This evaluation methodology based on the correlation between the strength scores and the resistance of passwords to attacks of increasing complexity can be regarded as a secondary contribution of the present work.

B. Experiment 2: Deterministic Evaluation

The statistical methodology presented in experiment 1 is a strict way of evaluating the performance of strength meters. Such statistical analysis allows determining if, from a general perspective, the method presents a consistent behaviour.

As a complement to the evaluation presented in experiment 1, studying the strength assigned to specific passwords can give additional insight into the strengths and limitations of a particular algorithm. With this objective, the multimodal meter was also computed for some individual password examples. The examples have been chosen so as to reflect some illustrative human behaviours in password selection in order to see the way in which the meter reacts to these changes. Common trends that have been taken into account for the selection of the individual examples are for instance: very common word-inspired password, inclusion of numbers, inclusion of special characters, inclusion of upper-case letters, making the password longer, selection of random passwords.

The selected password examples, together with their corresponding multimodal strength, are presented in Table IV. As in experiment 1, for comparison purposes, also the strength assigned by the NIST, Yahoo, Gmail and Dropbox appear in the following columns of the table.

1) Multimodal Meter: Even though this is just a particular deterministic example, the multimodal scores shown in the table raise some interesting points regarding the usual trends in human password selection and how these really affect the strength of the final password:

- Example 1 in Table IV is most often ranked in all datasets among the top 5 most used passwords. As such, it is present in any blacklist of banned passwords. Consequently, it is detected as a trivial password and is assigned a strength very close to 0.
- A large strength difference may be observed between passwords 1 (trivial very common password) and 2, which differ in just two characters. This length variation makes the second password resistant to brute-force attacks, which accounts for the strength increase.
  Please recall that, as mentioned in the training phase of the algorithm (see Sect. III), the length of brute-forceable passwords is a parameter that can be adjusted depending on the specific conditions of each application.
- Adding more characters (another known word) between passwords 2 and 3, significantly increases the strength level. Even if it only contains lower-case letters, password 3 is now length 15 and is the result of a combination of three words, which gives it a fair strength level.
- Passwords 3, 4, 5, 6 and 7 show that, for the same length 15, adding just *one* character of a different class (i.e., upper-case, digits, or special character), does not necessarily have a significant impact on the strength of the password: this depends heavily on the specific character added and the position in which it is added.
- Comparing passwords 3 and 8 it may be noticed that, for the same length, adding *several* characters of a different class (i.e., upper-case letters, password 8) to a password composed of just one-class characters (i.e., lower-case letters, password 3), has a positive impact in its strength.
- The comparison of passwords 8, 9 and 10, shows that, for the same length, adding characters of different new classes (i.e., digits and special characters, passwords 9 and 10) to a password composed of already two-class characters (i.e., lower- and upper-case letters, password 8), usually implies a moderate increase in the strength. As before, the significance of this increase

depends on the specific characters that are added and the position in which they are added.

- Password 11 shows that, as a quite consistent trend: in case of passwords originally based on known words, longer passwords are in general stronger than shorter passwords, even if the latter have more character classes.
- Finally, passwords 12 and 13 show that randomness is, together with length, the best policy to produce strong passwords. As it can be seen, password 12 is length 8 and contains only numbers and lower-case letters, however, its strength is comparable to password 10 which contains all character types and is length 15. Similarly, password 13 which is also length 8 but contains all character types has a strength equivalent to example 11 which is length 25.

Regarding the other four considered meters from the state of the art, different inconsistencies can be pointed out from their results in Table IV:

2) Nist Meter: It fails to give reasonable strength estimations for some of the examples. For instance, the initial password, which is probably the first guess that any attacker would make, is given a medium strength value. Furthermore: a random password such as 12 is assigned the same value as "password"; another random password like 13 is assigned only a slightly higher value than a password with just lower-case letters composed of two known words such as "mypassword" (number 2). These observations reinforce the results from experiment 1: this meter has a tendency to overestimate weak passwords and to underestimate strong ones.

3) Yahoo Meter: The same as NIST, it overestimates the strength of the first password. Furthermore, password 1 is given the same strength as password 11, which is 25-character long. This is probably due the fact that both passwords contain only lower-case letters.

4) Gmail Meter: As seen in experiment 1, this meter tends to overestimate the strength of passwords. As the previous two meters, it does not provide a good estimation for password 1. Also, it does not seem reasonable to consider the same strength level for passwords 11-13, than for passwords 3-7.

5) Dropbox Meter: On the other side of the spectrum, Dropbox seems to underestimate the strength of passwords. It is very arguable whether "password" (which is rightly given the lowest strength level) should be assigned the same strength as "MyPassw0rdR0cks". Similarly, a 25-character password (number 11) is just given a medium strength.

The results of experiment 2 do not mean that the multimodal meter is flawless. Most likely, particular examples can be found for which it provides questionable strength estimations. Finding such counterexamples is one of the objectives of releasing the Multi-PaStMe application. However, together with experiment 1, it does show that the proposed meter is, in many ways, more reliable than other popular methods.

In a nutshell, the results reached in the two evaluation experiments give some more solid ground to confirm what is nowadays a widely held intuition among the password community: in order to select strong passwords, get random, get long, or, even better, get both.

VII. Discussion: The Use of Passwords and Strength Meters

The old debate still remains: Are passwords obsolete? The ever-lasting question has recently recovered all its strength following the series of cyber-attacks that have either been based on password guessing or that have led to password breaches. For the time being, what is clear is that passwords are, and will be, around for quite some time.

Password critics base their position on an argument difficult to refute: password security has many weaknesses, some of which are not difficult to be exploited. While this is certainly true, *security* is just one side of the coin. The other side is *convenience* [31]. If passwords are so bad as authentication means, why are they so stubbornly resilient? Why haven't they been replaced long ago by any of the other existing authentication methods? The explanation to the their popularity is simple: they are easy to implement by service providers and they are easy to utilize by end-users. They may not be very secure, but they certainly are convenient.

Is there an over-use of passwords? Probably so. If security is a must, other type of authentication methods should be used. For instance, in recent years, the banking sector has widely deployed multifactor authentication for their online services in response to the growing number of cyber-attacks targeting end-users and leading to big economic losses [32].

Should we then forget about passwords? Probably not. It is doubtful whether a high-security multifactor authentication is the best solution in order to login to our favourite on-line cooking blog. A simple password like "cupcakes" will most likely do the job. Will we forget it? Certainly not. Is it secure? Neither. But then again, in this case, are we looking for high convenience or for high security?

In any case, a common and advisable practice is to guide users in the process of selecting a password at the time of registration to a service. Users should be informed regarding the strength of the password they select and the risks associated to it. If this feedback is reliable, it is then the user's responsibility to decide the type of password he wants to select in order to protect: A) his on-line bank account; B) his cooking recipes.

The key concept in the previous discussion is: *reliable feedback*. To date, such a feedback has been traditionally supported by: 1) password composition policies [33]; or 2) password strength meters [7].

Traditional password composition policies oblige users to observe certain requirements in most cases related to the length and/or complexity of their passwords [33]. These rules are thought to increase the password space and are therefore an efficient protection against brute-force attacks (similar to module 1A of the multimodal approach). In spite of their wide use, different studies have pointed out some of the general limitations that password composition policies present in different degrees [3], [34]–[36]: 1) in many cases they are regarded as annoying by end-users and 2) they do not necessarily lead to stronger passwords. For example, a password composition policy that requires the use of at least a capital letter and a number would reject a random password of 20 lower-case characters. Furthermore, certain policies can even be counterproductive, by inducing users to follow certain patterns that can later be exploited by guessing attacks. For instance, it is true that forcing the user to include a capital letter in the password increases substantially the password space to be searched by brute-force attacks. However, it is also true that most users will include the capital letter at the beginning of the password if they are not given any other directive. Dictionary attacks can take advantage of this known behavior by simply adding the rule "capitalize first letter".

Given the limitations of rule-based password composition policies [37], password strength meters have emerged as an alternative to help users in their password selection [8]. Password strength meters can be used to enforce password policies or simply to advice users allowing them a higher degree of freedom regarding the password they choose, without constraining them to follow any specific pattern. As feedback, the user receives a strength estimation of the selected password. That is, meters are designed to help users understand if their password choices will resist attempts to crack them. Although the core idea behind is sound, the problem of most current strength meters is their lack of reliability [6]. This is a big problem, as a bad password strength meter can be worse than useless: it can induce users to believe that they are selecting strong passwords when they are not.

So, password strength meters can be a powerful tool in password selection, as long as they truly serve their purpose: providing *reliable* feedback to the user. The experiments provided in the present article have shown precisely that: the new proposed multimodal approach is a reliable estimator of password strength, improving the output given by other largely used methods. Furthermore, it is aligned with the new draft of the NIST recommendation [38], that clearly changes the philosophy of previous versions towards new more comprehensive guidelines not only focused on rules.

Regarding the practical use of the proposed meter, different studies on strength meters have shown that users are more responsive to word-based feedback, rather than numerical values that are difficult to interpret [2], [8]. This is the case, for instance, of the three commercial strength meters analysed in the experimental sections (i.e., yahoo, gmail and dropbox) which consider a ranking of 4-5 different strength levels such as: 'very weak', 'weak', 'so-so', 'good' and 'strong'. Following this good practice, the multimodal score range (0-10) could be divided into a fixed number of strength levels.

VIII. Conclusions

As described in Part I of this series of two papers [1], the new multimodal strength meter was initially devised following:

1) The general principle "strength in numbers" which, for this particular case, can be phrased as: do not dismiss certain imperfect password strength approaches leaving all the responsibility of correctly estimating the robustness of passwords to just one algorithm. Instead, combine different methods specialized in detecting a particular group of weak passwords to generate a more general and reliable overall approach.
2) The "complementarity principle" from the information fusion field: the combination of complementary systems measuring different properties of the same problem tends to provide better results than the individual algorithms by themselves. We refer the reader to Annex B, provided as accompanying material of the present paper, for further discussion on the complementarity of the two Markov-based modules of the multimodal meter.

The evaluation presented in the current paper has proven the sensibility of the approach and how, based on both principles, the final multimodal method is capable of outperforming largely used meters from the state of the art.

As a first phase, the algorithms has been trained in order to adjust it to a quite typical environment: English-based application, not designed for a specific community but thought for generic users (e.g., online email), vulnerable to offline attacks, using a simple hashing algorithm such as MD5. After that initial training phase, the algorithm has been evaluated following an innovative experimental framework including a three-dimensional evaluation: statistical, deterministic and third parties public comparison. The results from these experiments produced the next conclusions:

- Experiment 1 has shown that, from a statistical perspective, the algorithm presents a high correlation between the complexity of the attack that broke a given password and the strength assigned to that password.
- Experiment 2 supported the claim that, also when considering specific password examples, the algorithm gives sensible strength estimations.
- Experiments 1 and 2 combined have assessed the high reliability and consistency of the strength estimations provided by the proposed multimodal method compared to other popular meters from the state of the art.

It should also be highlighted as a salient feature of the proposed method its high flexibility. It can be adjusted during the training phase to work on application specific environments or adapted to a certain password policy. Such capability is the key to overcoming one of the traditional shortcomings of previous strength meters, which would assign the same strength to a given password independently of the context where it was used. By exploiting this adaptability, the meter can also evolve over time in order to naturally adjust to new password selection trends. This could be the case, for instance, of a certain community of users that, driven by a given password policy, begins to systematically choose passwords with a predictable structure that could be exploited by an attacker [34], [35]. In this situation, the model would be able to adapt to the new trend and start rating such passwords as low security ones. This adjustment process would be accomplished by retraining the probability matrices on new data representing the variability of that particular population.

As a wrap up conclusion of the present series of two works, it may be stated that, in spite of some dooming predictions regarding their future [39], passwords are still the most commonly used method of web-based personal authentication and it is not likely that they will be replaced in the coming years [40]. Therefore, it is not enough to just blame the users for potential security breaches, but we should focus on devising new methods to help them choose better and stronger passwords. Hopefully, works such as the present one can help to move forward in the path towards this goal.

REFERENCES

[1] J. Galbally, I. Coisel, and I. Sanchez, “A new multimodal approach for password strength estimation. Part I: Theory and algorithms,” *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 2829–2844, Dec. 2017.

[2] S. Furnell, “Assessing password guidance and enforcement on leading Websites,” *Comput. Fraud Secur.*, vol. 12, no. 12, pp. 10–18, 2011.

[3] M. Weir, S. Aggarwal, M. Collins, and H. Stern, “Testing metrics for password creation policies by attacking large sets of revealed passwords,” in *Proc. ACM CCS*, 2010, pp. 162–175.

[4] C. Castelluccia, M. Dürmuth, and D. Perito, “Adaptive password-strength meters from Markov models,” in *Proc. NDSS*, 2012. [Online]. Available: https://www.internetsociety.org/doc/password-creation-presence-blacklists

[5] X. de C. de Carnavalet and M. Mannan, “From very weak to very strong: Analyzing password-strength meters,” in *Proc. NDSS*, 2014, pp. 23–26.

[6] X. de C. de Carnavalet and M. Mannan, “A large-scale evaluation of high-impact password strength meters,” *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 1, pp. A-1–A-31, 2015.

[7] B. Ur *et al.*, “Helping users create better passwords,” in *Proc. LOGIN*, vol. 37. 2012, pp. 51–57.

[8] B. Ur *et al.*, “How does your password measure up? The effect of strength meters on password creation,” in *Proc. USENIX Secur. Symp.*, 2012, pp. 65–80.

[9] H. Habib *et al.*, “Password creation in the presence of blacklists,” in *Proc. NDSS WUSEC*, 2017.

[10] WhatsMyPass. (2008). *The Top 500 Worst Passwords of All Time*. [Online]. Available: http://www.whatsmypass.com/the-top-500-worst-passwords-of-all-time

[11] TechCrunch. (2009). *370 Passwords You Shouldn’t (and Can’t) Use on Twitter*. [Online]. Available: http://techcrunch.com/2009/12/27/twitter-banned-passwords/

[12] ReusableSecurity. (2009). *The RockYou 32 Million Password List Top 100*. [Online]. Available: http://reusablesec.blogspot.it/2009/12/rockyou-32-million-password-list%-top.html

[13] D. Florencio, C. Herley, and P. C. Van Oorschot, “An administrator’s guide to internet password research,” in *Proc. USENIX LISA*, 2014, pp. 35–52.

[14] R. Shay *et al.*, “Designing password policies for strength and usability,” *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 4, pp. 13-1–13-34, 2016.

[15] KoreLogic. (2016). *KoreLogic Public Passwords Dataset*. [Online]. Available: https://www.korelogic.com/InfoSecSouthwest2012_Ripe_Hashes.html

[16] m3g9tron. (2016). *Cracking Story—How I Cracked Over 122 Million SHA1 and MD5 Hashed Passwords*. [Online]. Available: https://blog.thireus.com

[17] W. E. Burr *et al.*, “NIST special publication 800-63-2: Electronic authentication guideline,” NIST, Gaithersburg, MD, USA, Tech. Rep. SP 800-63-2, 2012.

[18] S. Ji, S. Yang, T. Wang, C. Liu, W.-H. Lee, and R. Beyah, “PARS: A uniform and open-source password analysis and research system,” in *Proc. ACM ACSAC*, 2015, pp. 321–330.

[19] R. Veras, C. Collins, and J. Thorpe, “On the semantic patterns of passwords and their security impact,” in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014. [Online]. Available: https://www.internetsociety.org/doc/semantic-patterns-passwords-and-their-security-impact

[20] J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic password models,” in *Proc. IEEE SSP*, May 2014, pp. 689–704.

[21] D. Jaeger, C. Pelchen, H. Graupner, F. Cheng, and C. Meinel, “Analysis of publicly leaked credentials and the long story of password (Re-) use,” in *Proc. Int. Conf. Passwords*, 2016.

[22] M. Weir and S. Aggarwal, “Cracking 400,000 passwords or how to explain to your roommate why the power-bill is a little high,” in *Proc. DefCon Conf.*, 2009.

[23] B. Ur *et al.*, “Measuring real-world accuracies and biases in modeling password guessability,” in *Proc. USENIX Secur. Symp.*, 2015, pp. 463–481.

[24] J. Steube, “Introducing the PRINCE attack mode,” in *Proc. Int. Conf. Passwords (PASSWORDS)*, 2014.

[25] OpenWall. (2016). *John-the-Ripper Open Source Password Cracker*. [Online]. Available: http://www.openwall.com/john/

[26] CrackStation. (2016). *CrackStation’s Password Cracking Dictionary*. [Online]. Available: https://crackstation.net/

[27] KoreLogic. (2016). *KoreLogic Set of Rules*. [Online]. Available: http://contest-2010.korelogic.com/rules.html

[28] D. L. Wheeler, “zxcvbn: Low-budget password strength estimation,” in *Proc. USENIX Secur. Symp.*, 2016, pp. 157–173.

[29] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[30] J. Bonneau, “The science of guessing: Analyzing an anonymized corpus of 70 million passwords,” in *Proc. IEEE SSP*, May 2012, pp. 538–552.

[31] L. Tam, M. Glassman, and M. Vandenwauver, “The psychology of password management: A tradeoff between security and convenience,” *J. Behaviour Inf. Technol.*, vol. 29, pp. 233–244, 2010.

[32] Symantec. (2016). *Dridex: Financial Trojan Aggressively Spread in Millions of Spam Emails Each Day*. [Online]. Available: https://www.symantec.com

[33] J. Campbell, D. Kleeman, and W. Ma, “The good and not so good of enforcing password composition rules,” *Inf. Syst. Secur.*, vol. 16, no. 1, pp. 2–8, 2007.

[34] S. Komanduri *et al.*, “Of passwords and people: Measuring the effect of password-composition policies,” in *Proc. CHFCS*, 2011, pp. 2595–2604.

[35] J. Campbell, W. Ma, and D. Kleeman, “Impact of restrictive composition policy on user password choices,” *Behaviour, Inf. Technol.*, vol. 30, no. 3, pp. 379–388, 2011.

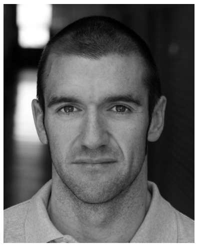
[36] R. Shay *et al.*, “Designing password policies for strength and usability,” *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 4, pp. 13-1–13-34, 2016.

[37] B. Ur *et al.*, “‘I added ’!’ at the end to make it secure’: Observing password creation in the lab,” in *Proc. USENIX SOUPS*, 2015, pp. 123–140.

[38] NIST. (Aug. 2016). *NIST SP 800-63-3: Public Preview*. [Online]. Available: https://pages.nist.gov/800-63-3/

[39] L. St. Clair *et al.*, “Password exhaustion: Predicting the end of password usefulness,” in *Proc. ICISS*, vol. LNCS-4332. 2006, pp. 37–55.

[40] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, “The quest to replace passwords: A framework for comparative evaluation of Web authentication schemes,” in *Proc. IEEE SSP*, May 2012, pp. 553–567.



Javier Galbally received the Ph.D. degree in electrical engineering from the Universidad Autónoma de Madrid, Spain, in 2009. In 2013, he joined the European Commission in the DG Joint Research Centre, where he is currently a Scientific Project Officer. His research interests are mainly focused on pattern and biometric recognition. Recently, he has started applying his knowledge in machine learning to password related problems.

Iwen Coisel received the Ph.D. degree in cryptography from Orange Labs, France, in 2009. He has been a Scientific Project Officer since 2012 with the Joint Research Centre of the European Commission, Ispra, Italy. Before joining the European Commission, he was a researcher with the Crypto Group of the Université Catholique de Louvain, Belgium, where he focused on private authentication systems.

Ignacio Sanchez received the M.Sc. degree in computer engineering from the University of Deusto, Spain, and the Ph.D. degree in computer engineering from the Spanish National University for Distance Education, Spain. He has 14 years of experience in the domain of information security. He is currently with the DG Joint Research Centre of the European Commission, where he is currently leading several research projects in the field of cybersecurity, privacy, and data protection.