Using Markov Models to Crack Passwords

Renier van Heerden
Council for Scientific and Industrial Research, South Africa

Johannes S Vorster
University of the Western Cape

Using Markov Models to Crack Passwords

R. P. van Heerden and J.S. Vorster
DPSS, CSIR, Pretoria, South Africa
rvheerden@csir.co.za
jvoster@csir.co.za

Abstract: We present a Markov Model for cracking and measuring quality of passwords. The Markov Model represents the transitions between specific characters. The Markov Model was built from a list of captured passwords, thus generating a password model with the frequency of passwords also incorporated. Traditional password quality measurement tests only against large dictionaries. We found that through the Markov Model character transition map we can optimise the search sequence for partially known passwords.

Keywords: Markov Model, password cracking

1. Introduction

Passwords are part of everyday life. People choose passwords according to schemes and methods which can be modelled. With a password model, our capability to crack passwords and measure password strengths can be enhanced.

For a password to be considered strong the following rules were suggested (Gehringer 2002) (Garfinkel 1991)

- Upper and Lower Case to be used
- Numerical symbols to be included
- Length of eight or more characters required
- No dictionary words to be used
- No names to be used
- Easy to remember

The last rule is the one that most computer users use. The rules were designed to defeat brute force and dictionary attacks. Human nature will still tend to choose passwords which are easy to remember. Florencio and Herly (2007) concluded that a large number of passwords used are of poor quality and are re-used.

“Asdf1234” comply with the above password rules. It will be difficult to crack with a brute force attack, and is not listed in dictionary. Possible patterns that people use are:

- Start with a capital letter
- Follow keyboard sequences
- End with numerical symbols

These patterns may not be as obvious as listed above. By using real passwords, we can map password patterns in a Markov Model.

2. Password Background

Many of the deficiencies of password authentication systems arise from human memory limitations (Yan 2004). A study done by Cambridge University to test the trade-off between security and memorability yielded the following results out of 288 test cases (Yan 2004):

- The average passwords were between 7 and 8 characters,
- Password selection advice was ignored by some users when not enforced,
- Permutations of dictionary words and numbers are popular,
- Special characters use was very limited.

Out of this study we learned that the memorability of passwords is the main factor in password choice.

3. Related Research

A connectionist Password Quality Tester was developed by Duffy and Jagota (Dyffy 2002). They developed a neural network system to test the quality of passwords. Their system was designed to discover which passwords become spurious. The traditional method of testing for spuriousness is to compare passwords to a large dictionary (Bishop 1995).

Ru and Eloff (du Ru 1997) looked at a methodology that uses fuzzy logic to perform typing biometrics. They found that users' fingers became accustomed faster to strings based on "English-like" text. Thus, users passwords choices are also influenced by the typing biometrics.
A method called "Markov filtering"(Natayanan 2005) which use a *zero* and *first* order Markov models was used to look at the possible time saving that can be achieved. Natayanan (2005) could recover 67% of passwords using a 2 x 109 search space which is a higher percentage than rainbow attacks.

4. Markov Model background

A Markov Model is a sequence of events - usually called states – for which the probability is dependent only on the event immediately preceding it (Rabiner 1989). That is only on the single previous event, not the history of events. Only the preceding event is of importance. As an example, if we now that the previous event was E, then the probability that the next event will be A is based only on E. Thus the E-A combination forms a unity and a probability can be assigned to it.

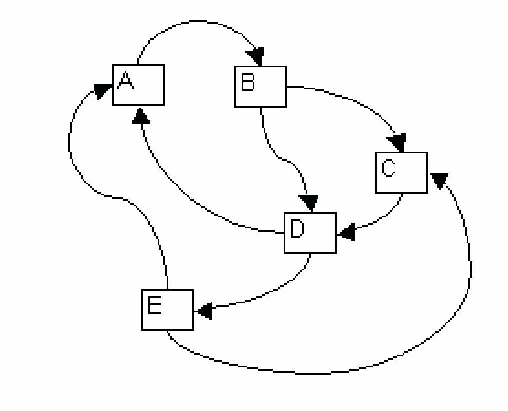

Figure 1: Markov Model

Above is a diagram that represents a number of letters and their relation to each other. In this diagram the arrows indicate a is-followed-by relation. Thus, A is followed by B, D is followed by E and so on.

Some letters are followed by more than one letter. This indicates an "OR", that is B is followed by D or B is followed by C. Using this diagram one can now construct "words" where a word is a number of letters following each other. For example the word "ABDECD" can be formed by following the arrows. In this context some words can be formed – called valid words, and others cannot be formed – called invalid words.

An example of an invalid word would be "ABDECC", because C MUST be followed by a D, and cannot be followed by another C. Such a model is often called a finite state model.

A Markov model is an extension from the finite state model. In a Markov model the transitions - or arrow - represents the probability that that transition will occur. It is possible to modify the above finite state model in such way that it becomes a Markov model. This can be done by adding probabilities to the transitions.

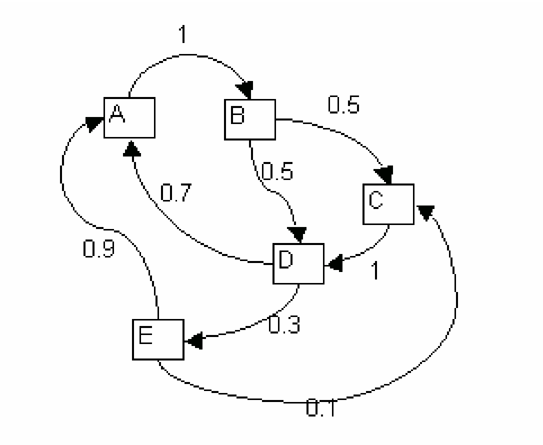

Figure 2: Markov Model with Probabilities

From this it can be seen that the probability that a B will follow an A, also written P(A → B) = 1.0 and P(B → C) = 0.5 and so on.

Because these are probabilities, the sum of all outgoing transitions must add up to one. There are two transitions from E: E to A and E to C. The probabilities of these are 0.9 and 0.1 respectively. These probabilities add up to 1.0.

It is possible to construct a Markov model of common passwords by interpreting each letter in a password as a state – or event – and then the probability of the transitions are given by the probability that some letter follows another.

5. Simple Markov Model

Multiple Markov Model architectures were used. The first model we used was similar to the one used in Figure 1. In Figure 3 we show a simplified version of the architecture with only 3 states, A, B, and C. The model has 223 states, representing all the available ASCII characters from 0x21 to 0xFF. 0x00 to 0x20 are used for print control and thus not available for password use (Norton 1986) .

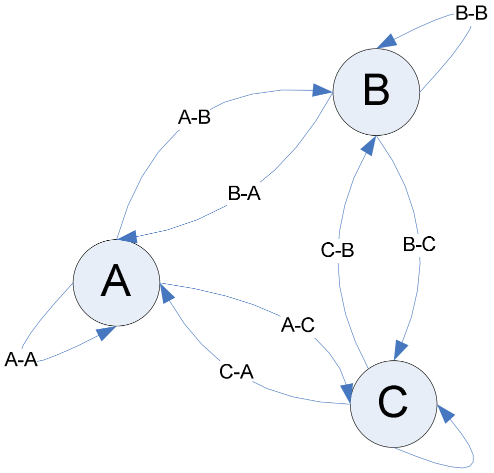

Figure 3: Simple Markov Model

In Figure 3 the transition between state A to state B is nominated as "A-B". The model in Figure 3 will only map passwords that consist of the A, B and C symbols. The password "AABCAA" will be mapped as follows:

Table 1: Password "AABCAA"

State	Transition
A	A-A
A	A-B

B	B-C
C	C-A
A	A-A
A	…

Each password added to the model will change the probabilities of the state transitions with the following limitations:
P(A-A) + P(A-B) + P(A-C) =1,
P(B-A) + P(B-B) + P(B-C) =1,
P(C-A) + P(C-B) + P(C-C) =1,

6. Start End Markov Model

A small variation on the model used in Figure 3 can be created by adding two more states. The states are “Start” and “Stop”. These new states model the transition to and end of a password. With this model frequent use of starting capitals and end use of numbers can be modelled. A simplified version of the Start End Markov Model is shown in Figure 4.
The “ABCCA” password transitions will change to:

Table 2: Password “ABCCA”

State	Transition
Start	Start-A
A	A-A
A	A-B
B	B-C
C	C-A
A	A-A
A	A-End
End	

With this model we are capable of mapping to map the popular choice of using capitals to start passwords.

7. Symbol Number Markov Model

A layer representing each symbol position was added, thus increasing the model complicity. The state A was replaced by states A1, A2, ... A20. We decided on a maximum password length of 20 characters to cover our password list. The required password length has changed from 7 to 15 characters, although most users use passwords that are less than 7 characters (Clair 2006). For different password lists the number of states per character may have to be increased.

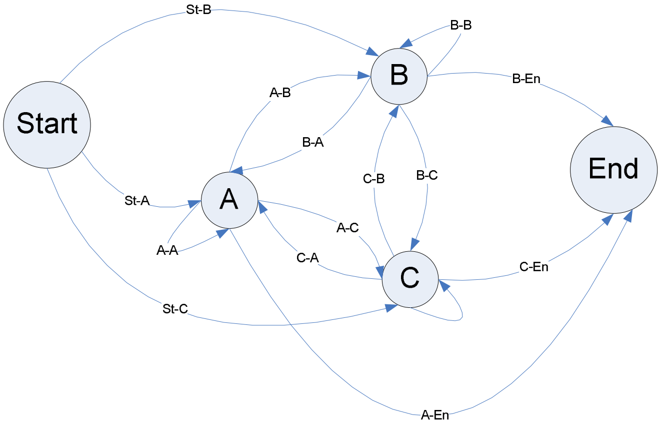

Figure 4: Start-End Markov Model

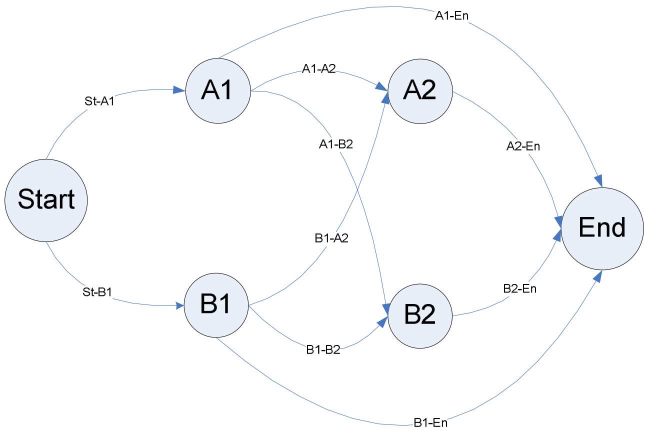

Figure 5: Symbol Number Markov Model

A simplified version is shown in Figure 5. The model in Figure 5 uses only two symbols: A and B, and limits the sequence number to 2. Thus the only possible passwords modelled in Figure 5 will be:

- A (Start-A1-End)
- AA (Start-A1-A2-End)
- AB (Start-A1-B2-End)
- B (Start-B1-End)
- BA (Start-B1-A2-End)
- BB (Start-B1-B2-End)

With this model we expect to map the use of numerals at the end of passwords.

8. Experiments

In our first experiment we calculated the Markov Model transitions with the highest occurrences. We used three different data sets:

- Dictionary (English 53 142 words),
- Real Password List (28 669 words),
- Generated Near Word List (1 785 719 words)

We listed the top 5 transitions for Simple and Start End Markov Models (Note the Real Password List was compiled from Windows NT machines and thus case insensitive)

Table 3: Simple Markov Model with Dictionary

States	#	% of #	Probability of Transition
i - n	9226	2.47	26.34
e - r	8683	2.33	20.77
e - s	6435	1.73	15.39
t - i	6141	1.65	23.35
t - e	5904	1.58	22.45

Table 4: Start End Markov Model with Dictionary

States	#	% of #	Probability of Transition
s - (End)	15051	3.14	42.74
i - n	9226	1.93	26.19
e - r	8683	1.81	17.94
(Start) - e	8069	1.68	15.19
(Start) – o	7328	1.53	13.8

Table 5: Simple Markov Model with Real Password List

States	#	% of #	Probability of Transition
A - N	2412	1.58	18.23
E - R	1946	1.28	15.41
0 - 0	1862	1.22	33.5
O - O	1837	1.00	16.59
M - A	1526	0.93	30.73

Table 6: Start End Markov Model with Real Password List

Start State – End State	#	% of #	Probability of Transition
(Start) - A	4268	2.04	14.95
(Start) - O	3483	1.66	12.2
(Start) - E	3105	1.48	10.88
E - (End)	2444	1.17	16.22
A -N	2414	1.15	15.94

Table 7: Simple Markov Model with Generated Near Word List

States	#	% of #	Probability of Transition
e – r	375907	2.67	22.21
e - n	320447	2.28	18.93
t – e	235122	1.67	27.35
i - n	198746	1.41	17.22
r – e	184485	1.38	19.17

Table 8: Start End Markov Model with Generated Near Word List

States	#	% of #	Probability of Transition
e - r	375907	2.14	19.42
e - n	320447	1.83	16.55
(Start) - e	281276	1.60	16.13
e - (End)	243310	1.39	12.57
(Start) -a	239975	1.37	13.76

The following conclusions can be made from Table 3 to 8:

- The “e-r” transition is very likely
- Most passwords start with “E”, “O” or “A”
- Most passwords ends with an “e”

In our second experiment we calculated the top transitions for the Symbol Number Markov Model. We listed only the first three symbol numbers and their top four occurrences. We used the dictionary and Real Password Lists.

Table 9: Symbol Number Markov Model with the Dictionary Password List

States	State Transition	#	% of #	Probability of Transition
Start - e	0 - 1	8069	7.59	15.19
Start - o	0 - 1	7328	6.89	13.8
Start - a	0 - 1	7552	6.89	13.65
Start - i	0 - 1	5020	6.82	9.54

o - n	1 - 2	1345	0.84	18.32
a - r	1 - 2	1306	0.82	17.97
r - a	1 - 2	1034	0.67	22.75
r - o	1 - 2	986	0.65	21.81
s - t	2 - 3	976	0.45	25.83
t - e	2 - 3	836	0.39	22.15
l - l	2 - 3	779	0.36	22.53
a -n	2 - 3	743	0.35	15.2

Table 10: Symbol Number Markov Model with the Real Password List

States	State	#	% of #	Probability of Transition
Start - A	0 - 1	4268	7.38	14.95
Start - O	0 - 1	3483	6.02	12.2
Start - E	0 - 1	3105	5.37	10.88
Start - I	0 - 1	2148	3.71	7.52
A-N	1 - 2	657	0.75	15.28
A-R	1 - 2	585	0.67	13.6
E-R	1 - 2	432	0.49	13.57
O-O	1 - 2	425	0.49	12.15
A-N	2 - 3	437	0.38	21.35
1-0	2 - 3	413	0.36	53.15
O-O	2 - 3	304	0.26	17.2
L-L	2 - 3	264	0.23	18.31

The following conclusions can be made from Table 9 and 10:

- Most passwords start with “E”, “O”, “I” or “A”
- The “A-R” transition is very popular between the second and third character
- If the third character is a “0”, there is a more than 50% chance that the next character is a “1”

In the next experiment we looked for differences in languages. We compared English, Dutch and Swahili.

Table 11: English, Dutch and Swahili Simple Markov Models

Ranking	English	Dutch	Swahili
1	i - n	e – n	w – a
2	e - r	e – r	l – i
3	e - s	g – e	k – a
4	t - i	d – e	a – u
5	t - e	t – e	k -u

In Table 11 it is apparent that the Markov Model differs for each language.

9. Practical example

One practical example where this model can be of use is when only part of a password is known. One of the simplest methods to gain a password is by overlooking the user while it is being typed. For example the following sequence was captured from a user which password is “HelloWorld”:

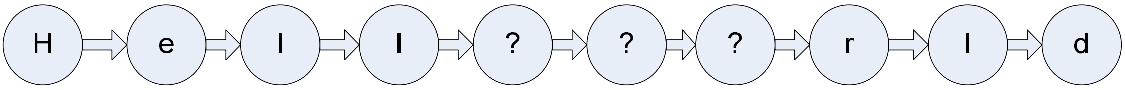

Figure 6: Overlook password

By combining two words, an automatic dictionary lookup will fail. While brute force system can usually work faster, it will start by testing “aaa” until “ZZZ”. This might not be optimum for a system that has a built in delay after 3 failed loggings. The sequence “Hello World” is commonly used by programming books as an example. Thus if the sequence “HelloWorld” has a high probability of being represented in a password collection. The model will also highlight the use of a capital “W” for the second word.

.

10. Future work

An optimal Markov Model structure for password modelling still has to be determined. The length of the character set can be increased to include sets of 2 or 3 characters.

Investigate the possible construct of a stochastic password cracker based on the probabilities of the Markov Model.

Mnemonic passwords (Kuo et al 2006) where a user choose a memorable phrase and use the first character to represent each word in the phrase can also be represented by this model. Where only 4% of mnemonic passwords are represented in dictionaries, a Markov model of memorable phrases can be build to represent this password scheme.

11. Conclusion

We employ a Markov Model built from known (cracked) passwords for two applications:

- Defensively, as a password strength evaluator
- Offensively, as a tool to enhance password guessing

We found that using Markov Models is a viable method for testing password strength because it assigns a probability to a password, and the inverse of this probability gives the number of password guesses that need to be employed to crack the password. Therefore one can use Markov Models to directly assign a number to a password that will indicate its ease of cracking (given the probability tables of previous passwords).

Secondly such a measurement gives an objective way to indicate to the user how strong her password is. This measurement is independent of dictionary word-lists. This is an improvement on password strength evaluators that use only dictionaries.

We found that Markov Models could also be used for improvements in brute-force password cracking. This can be achieved by using the transition probabilities to direct arrangement of letters in the construction of a password as a guess to the real password. Furthermore, such a Markov Model can be used for stochastic password cracking techniques.

References

Bishop, M. and Klein, D.V. (1995) *Improving system security via proactive password checking,* Computers and Security, vol. 14, no. 3, pp. 233–249

Clair, L.S. Johansen. L, Enck, W. Pirretti, M. Traynor, P. McDaniel, P. and Jaeger, T. (2006) *Password Exhaustion: Predicting the End of Password Usefulness,* Proceedings of 2nd International Conference on Information Systems Security (ICISS), pages 37--55

de Ru, W.G. and Eloff, J. h. P. (1997) *Enhanced Password Authentication through Fuzzy Logic,* IEEE Intelligent .Systems, Their Application pages 38-45

Dyffy, N. and Jagota, A. (2002) *Connectionist Password Quality Tester,* IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 4, pages 920-922

Florencio, D. and Herly C. (2007) *A large-scale study of web password habits*, Proceedings of the 16th international conference on World Wide Web, pages 657-666

Garfinkel, S. and Spafford, G. (1991) *Practical UNIX Security*, CA O'Reilly & Associates, Sebastopol, pp 35

Gehringer, E. F. (2002) *Choosing Passwords: Security and Human Factors,* Symposium on Technology and Society (ISTAS'02), pages 369-373

Kuo, C. Romanosky, S. and Cranor L.F. (2006) *Human selection of mnemonic phrase-based passwords*, Proceedings of the second symposium on Usable privacy and security, pages 67 - 78

Narayanan, A. and Shmatikov, V (2005) *Fast Dictionary Attacks on Passwords Using Time-Space Tradeoff*, Proc. of 12th ACM Conference on Computer and Communications Security (CCS)

Norton, P. (1986) *Inside the IBM PC*, Bradly p 41

Rabiner, L.R. (1989) *A tutorial on HMM and selected applications in speech recognition,* In Proc. IEEE, Vol. 77, No. 2, pp. 257-286

Yan, R., Blackwell,R., Anderson R. and Grant A. (2004) *Password Memorability and Security: Empirical Results,* IEEE Security and Privacy