Pass2Edit: A Multi-Step Generative Model for Guessing Edited Passwords

Ding Wang, **Yunkai Zou**
Nankai University
{wangding, zouyunkai}@nankai.edu.cn

Yuan-An Xiao
Peking University
xiaoyuanan@pku.edu.cn

Siqi Ma
The University of New South Wales
siqi.ma@unsw.edu.au

Xiaofeng Chen
Xidian University
xfchen@xidian.edu.cn

The 32nd USENIX Security Symposium

Passwords

Get from https://lorrie.cranor.org/blog/2013/08/12/security-blanket/

Passwords are irreplaceable

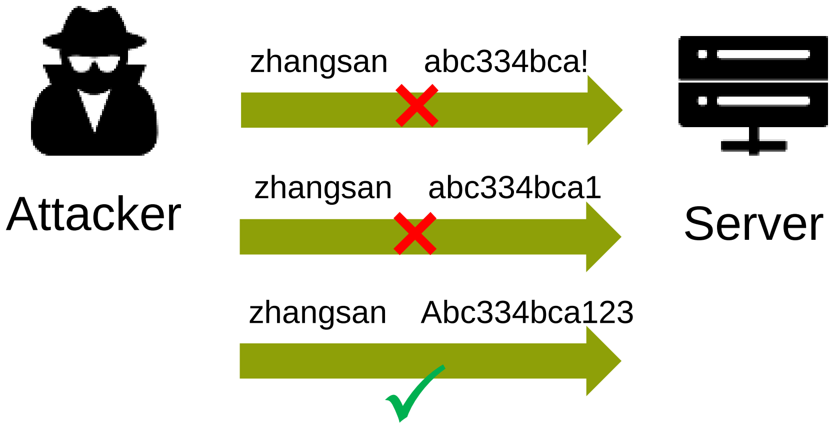
- Text passwords are **the most prevalent method** of user authentication.
- Other authentication technologies have fundamental flaws, and **passwords are irreplaceable** in the foreseeable future.

	Low cost	Useability	Renewability
Password	✓	Mid	✓
Hardware token	✗	Low	✓
Biometrics	✗	High	✗

Password reuse attack is realistic

- Typical Internet users are reported to have around **100 passwords** [1].
- 43%-51% of users **directly reuse** their existing passwords [2].
- **86%** of basic web application attacks were due to **stolen passwords.**【DBIR 2023】
- 21%-33% of users **slightly edit/modify** their existing passwords [3].

Username	Password
zhangsan	PW1:abc334bca …
…	…



Username	Password
zhangsan	PW2: Abc334bca123 ✓
...	...

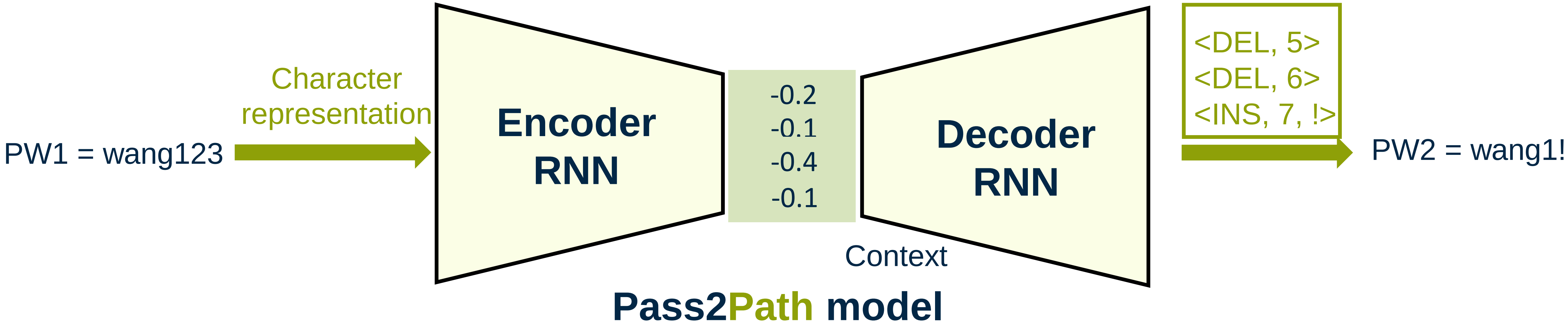
[1] https://tech.co/password-managers/how-many-passwords-average-person.
[2] The tangled web of password reuse. In Proc. NDSS 2014.
[3] Targeted online password guessing: An underestimated threat. In Proc. ACM CCS 2016.

Research on password reuse

Model	Type	Descriptions
Das et al. NDSS 2014	**Rule-based**	Eight **heuristic** transformation rules in **a predefined order**, e.g., deletion, insertion, reversal, etc.
Wang et al. ACM CCS 2016	**Probabilistic**	**PCFG-based** algorithm: **Two-step transformation** Structure-level transformation (e.g., $L_8D_3 \rightarrow L_8$) Segment-level transformation (e.g., 123456→12345)
Pal et al. IEEE S&P 2019	**Deep learning**	**Seq2Seq-based** model. Input: PW1 (e.g., 123456) Output: **the modification operation path** from PW1 to PW2 (e.g., 123456 → Delete 6 at the end)

Pal et al.'s Pass2Path model (IEEE S&P 2019)

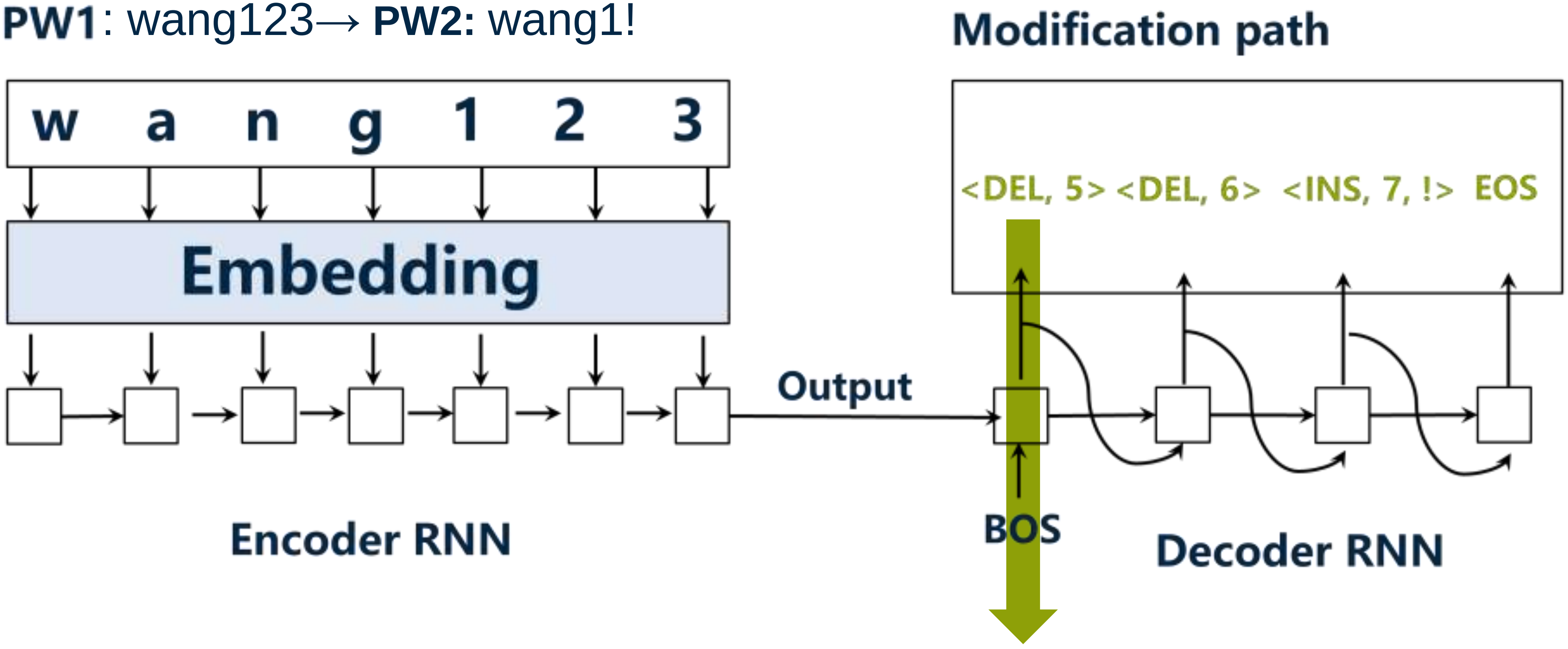
- Pass2Path defines **three character-level** atomic modifications: insertion, deletion, and substitution.
- Model input: user's old **password character sequence PW1**
- Model output: **a sequence of modifications** to transform PW1 to PW2.



Existing issues of Pass2Path (IEEE S&P 2019)

- Pass2Path cannot capture **the mutual influence** between password edit operations and corresponding transformation effects.

PW1: wang123→ **PW2:** wang1!



After the operation <DEL,5>, **wang123** **has already been modified to** **wang13**

Existing issues of Pass2Path (IEEE S&P 2019)

- Inaccurate similarity measurement

User	PW1	PW2
A	3080124	cooper3080124
B	720710	720710720710
C	wozuixiao	leizixi1
D	123456789	281456

✓ Reused pair
✓ Reused pair
× Non-reused pair
× Non-reused pair

Edit distance = 6

- Without consideration of popular passwords

User	PW1	PW2
Bob	abc334bca	12345678

PW2 is not similar to PW1

PW1 = abc334bca

Pass2Path

Guesses	Pr(PW2|PW1)
abc334bca1	0.6
abc334bca123	0.2
abc34	0.1
…	…

PW2 = 12345678 ×

Training data cleaning

- Password similarity metric: 2-gram cosine similarity > 0.3

PW1: abc→ [^a, ab, bc, c$]

PW2: abcabc → [^a, ab, bc, ca, ab, bc, c$] (^ and $ represent the **beginning and end symbols**)

	^a	**ab**	**bc**	**c$**	**ca**
abc	1	1	1	1	0
abcabc	1	2	2	1	1

sim(abc, abcabc) = cos< (1,1,1,1,0),(1,2,2,1,1)> = 0.905

- More accurate similarity measurement

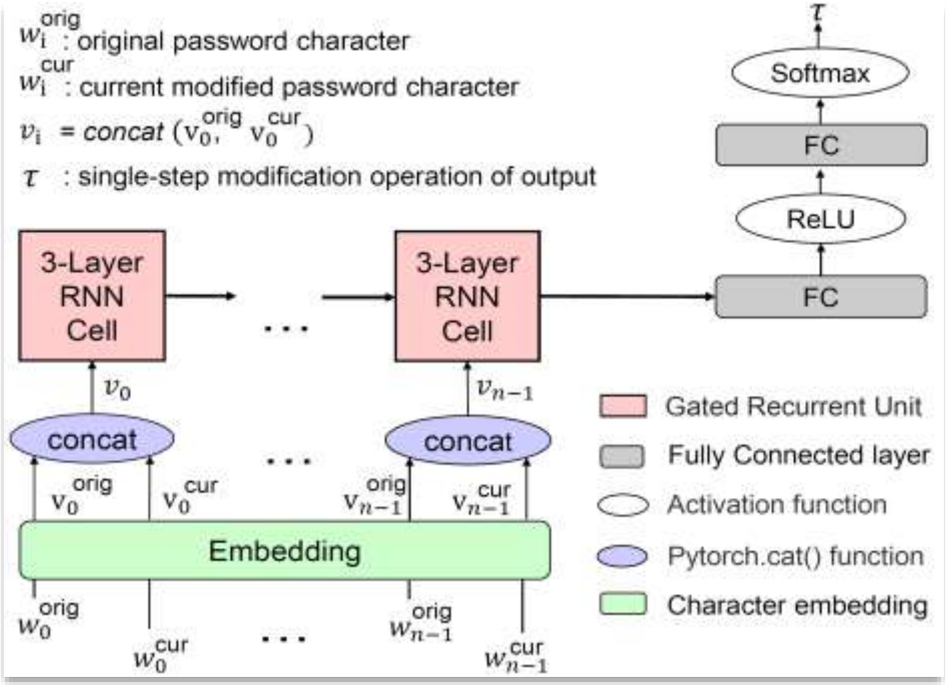
Users	**PW1**	**PW2**
A	3080124	cooper3080124
B	720710	720710720710
C	wozuixiao	leizixi1
D	123456789	281456

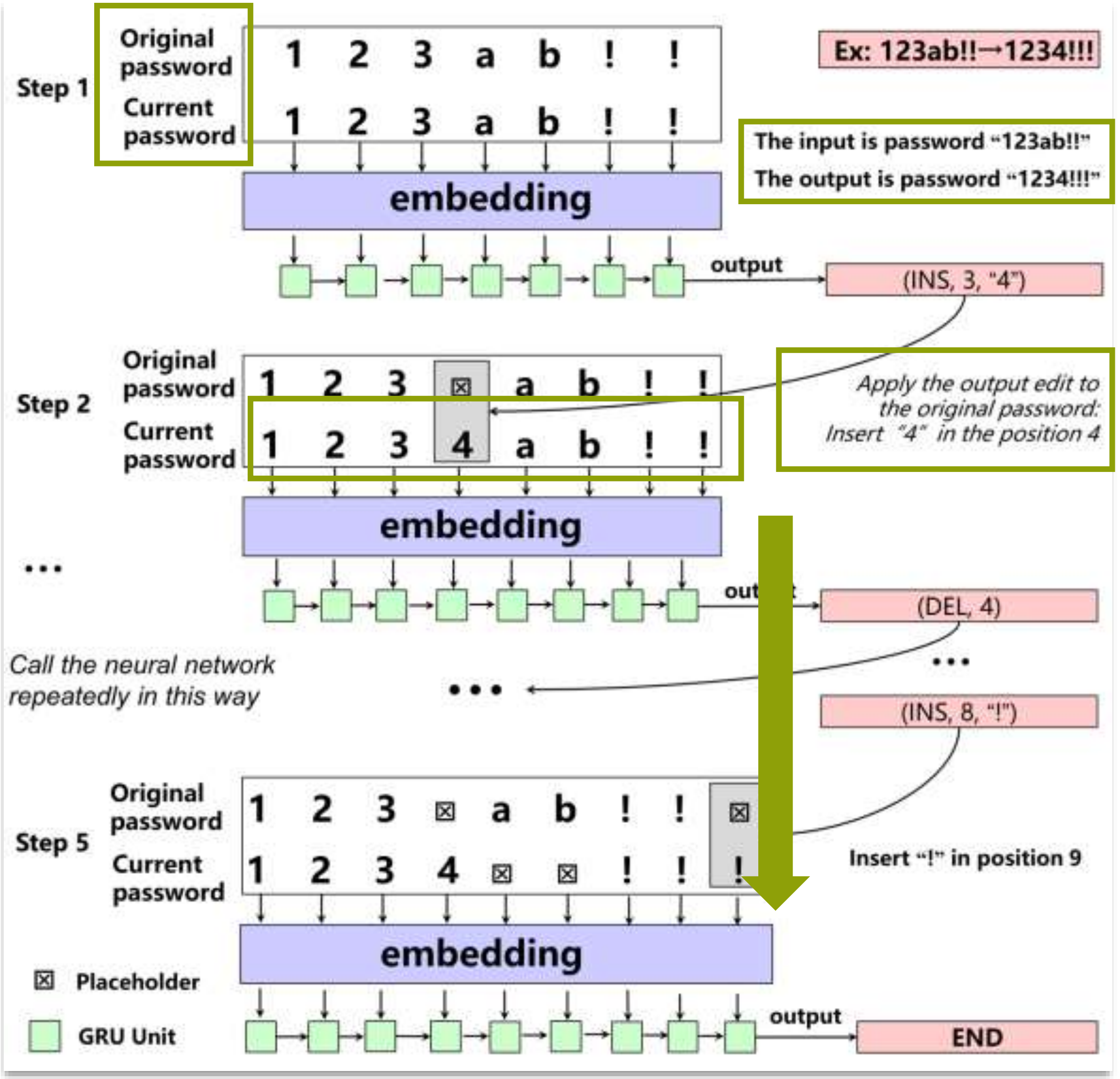
Users	**PW1**	**PW2**	**Similarity**	
A	3080124	cooper3080124	**0.66**	✓
B	720710	720710720710	**0.95**	✓
C	wozuixiao	leizixi1	0.21	✗
D	123456789	281456	0.24	✗

Pass2Edit: a multi-step generative model

Training process

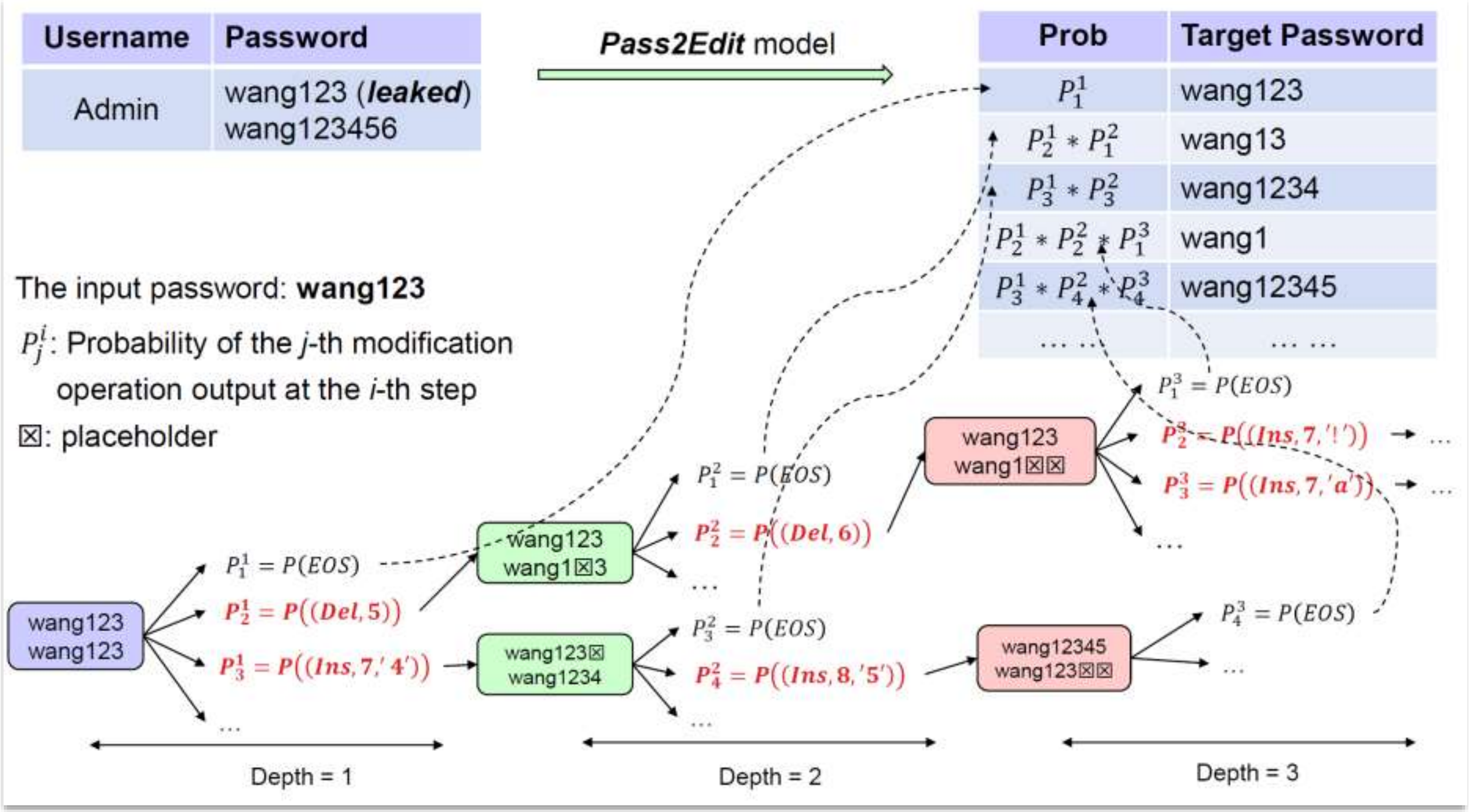
- The input at each step: the **original password** and the **current modified password**.
- The output at each step: **single-step modification operation.**





Password generation process

- Use the **beam search algorithm** to generate edited guesses.



Mixing popular passwords

- How to integrate **popular passwords**?
  - Multiply the probability of each **generated** password by a factor α.
  - Use **the frequency** of each popular password in the training set to estimate its probability.
  - Merge the two password sets **in descending order of probability**.

Beam Search Result

Output Prob (exponential)	Tweaked Prob (exponential)	Password
-2.39	-3.74	wang12
-2.71	-4.06	wang
-3.07	-4.42	wang1
-3.96	-5.31	123
…	…	…

Probability adjustment

Mixed Password List

Prob (exponential)	Password
-3.14	12345678
-3.28	123456789
-3.74	wang12
-4.06	wang
…	…

Sort after mixing

Popular Passwords

Prob (exponential)	Password
-3.14	12345678
-3.28	123456789
-4.41	11111111
-4.06	dearbook
…	…

From the training set

Experimental setup

- **Three research questions (RQs)**
  - **How well** does Pass2Edit perform?
  - How effective is our Pass2Edit **in practical attacking scenarios**?
  - Does **the efficiency** of our Pass2Edit meet the needs of the real attacker?

Table 2: Setups of 12 different attacking scenarios (RQ=Research question, see Section 4.2; For evaluation results, see Fig. 5)[†]

Scenario #	RQ# addressed	Language	Training set setup	Size (pairs)	Test set setup	Size (pairs)
1	RQ2	Chinese	Tianya → Dodonew	624,925	Tianya → Taobao	57,7017
2	RQ2		126 → Dodonew ($len \geq 8$)	188,926	126 → CSDN ($len \geq 8$)	85,206
3	RQ2, RQ3		CSDN → Dodonew	211,385	CSDN → 126	86,104
4	RQ2		Tianya → Dodonew ($len \geq 8$)	434,255	Tianya → CSDN ($len \geq 8$)	826,559
5	RQ2	English	000Webhost → Yahoo ($len \geq 6$)	265,083	000Webhost → LinkedIn ($len \geq 6$)	265,083
6	RQ2		Yahoo → LinkedIn (LD)	40,646	Yahoo → 000Webhost (LD)	37,479
7	RQ2		LinkedIn → Yahoo (LD, $len \geq 6$)*	40,812	LinkedIn → 000Webhost (LD, $len \geq 6$)	259,175
8	RQ1, RQ3	Mixed	80% of 3 mixed English datasets	338,857	20% of 3 mixed English Datasets	84,714
9	RQ1, RQ3		80% of 3 mixed Chinese datasets	434,255	20% of 3 mixed Chinese Datasets	108,564
10	RQ1, RQ3		80% of 4iQ dataset matched by email	116,837,808	20 % 4iQ dataset matched by email	29,209,452
11	RQ1, RQ3		80% of COMB dataset matched by email	342,921,727	20 % COMB dataset matched by email	85,730,432
12 (real)	RQ2	English	000Webhost → Linkedin (LD $len \geq 6$)	213,697	000Webhost → RedMart (LD $len \geq 6$)	6,858

[†] $A \rightarrow B$ means that: A user's password at service A can be used by an attacker to help attack this user's account at service B.
*(LD, $len \geq 6$) means that we only use passwords that contain at least one digit and one letter, and have a minimum length of 6 in the dataset.

Experimental results

- Within 100 guesses, the guessing success rates of our Pass2Edit are **18.2%-33.0% higher** than its foremost counterparts.
- The **training time and password generation speed** of our Pass2Edit fully meets the needs of a realistic attacker.

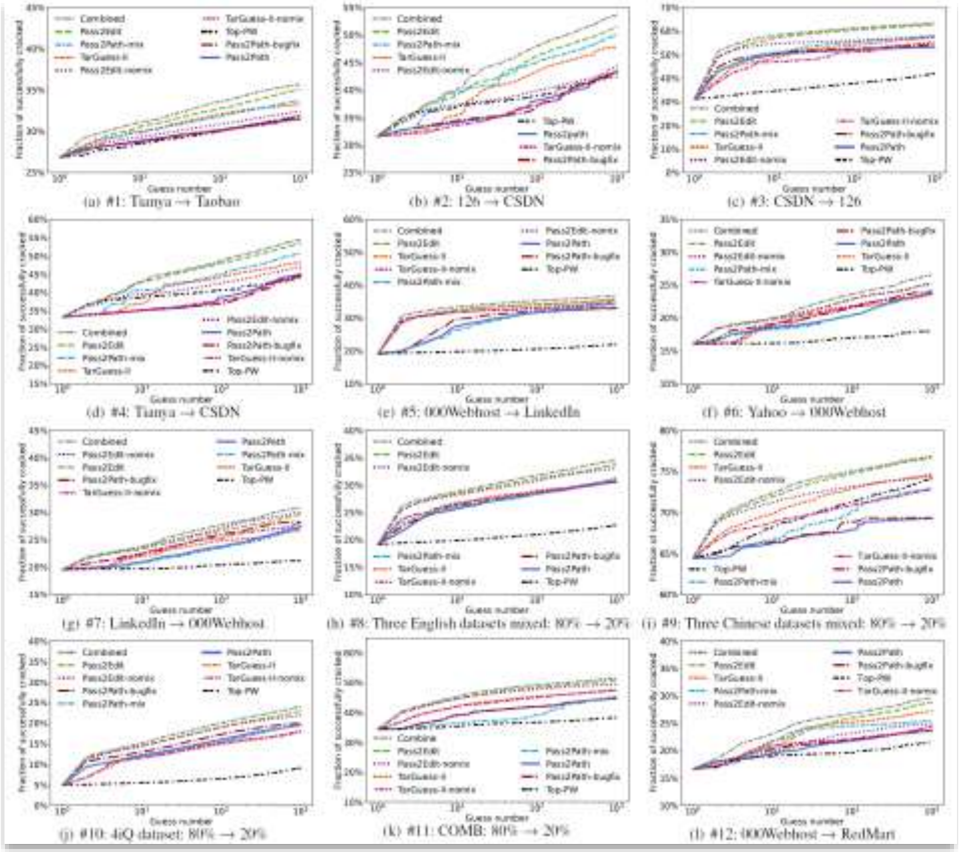
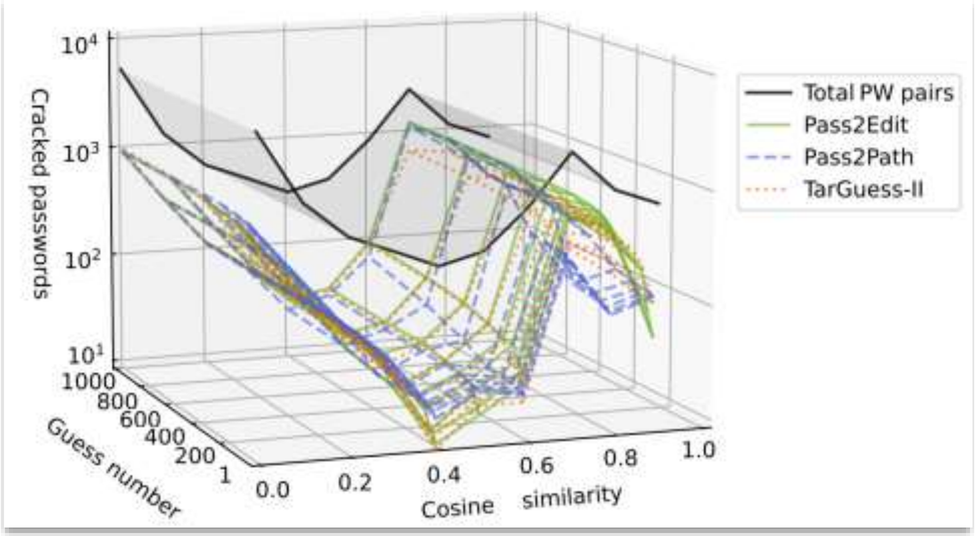


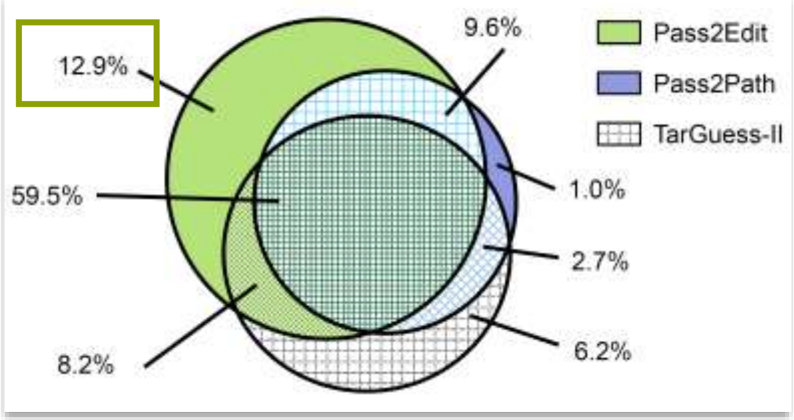
Table 6: Running time of different attack models.†

Attack method	Training time	Testing time	Generated PW/s ‡
TarGuess-II [71]	00:59:44	00:57:13	5,538
Pass2Path [46]	14:09:45	01:46:42	2,969
PASS2EDIT	09:43:26	02:26:25	2,164

† The timings are taken from attack scenario #10 and their format is "hour:minute:second". All model parameters are consistent with Sec. 4.3.
‡ PW/s is calculated by dividing the total number by the total testing time.



Analysis of cracked passwords



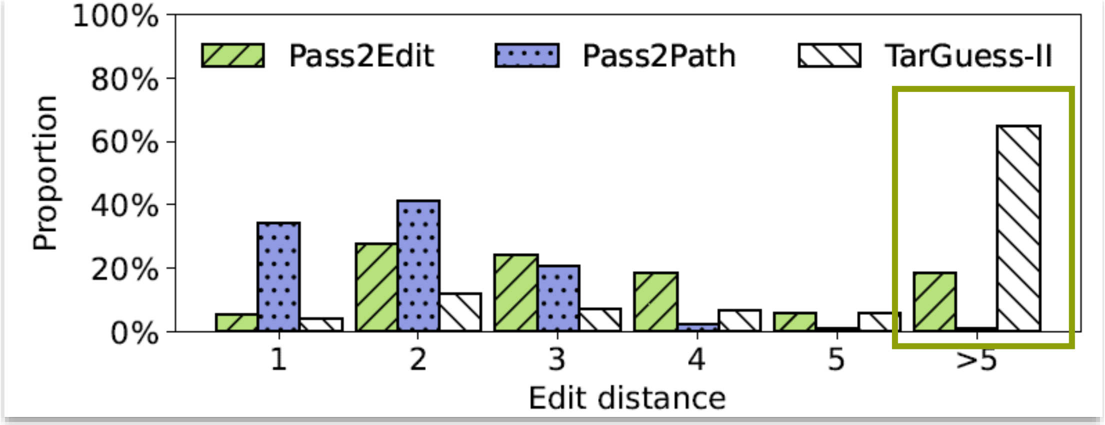


Table 8: Examples of passwords cracked independently by different models.

Attacking models		TarGuess-II [71]		Pass2Path [46]		Our Pass2Edit	
Number	Language	Existing password	Targeted password	Existing password	Targeted password	Existing password	Targeted password
1	Chinese	gxb840213	gxb1314521	biaokng	biaoking	201212	dai201212
2		dragonyr	123456789	ximmy851129	ximmy851119	9918241	zyj9918241
3		243586	qazwsxedc	199185	19910805	fire2500	ling2500
4		Tian6253*	love6253	zhangbig	ZHANGbig	1314520	1314520xl
5		2323kbc	123123kbc	super19771020	super19791020	6691064	6691064wu
6	English	seperti*	123456	JAtt12#$	JAtt1234	di10ca10040790	dica040790
7		sergioafull15013320	15013320	rajivamerica123	RAJIVamerica123	t@lking1	talking
8		megahomme@megahomme	megahomme	Iuliana93LAN	Iuliana93LaN	9427-078-168	9427078168
9		ddd786*1987	1987*786	kornjacica989	kornjaca89	Denningj11!!	denningj7
10		301873022iansangbbyboo	301873022	savone61	Savone6!	Ritalin!2#	ritalin123

Delete the letter segment

Takeaways and future work

- Employ Pass2Edit to generate **flat honeywords.**

Tiger03	tiger82	tiger59	tiger15	**tiger81**
tigeR17	tiger32	tiger8!	tiger70	Tiger88

- How to utilize **multiple existing passwords** of the same user to further improve the guessing success rate?

Username	Password
zhangsan	PW1:abc334bca PW2: password PW3: Abc334bca123 …
…	…

Username	Password
zhangsan	PWn: zhangAbc334
…	…

Thank you!

Pass2Edit: A Multi-Step Generative Model for Guessing Edited Passwords

Ding Wang, Yunkai Zou
Nankai University
{wangding, zouyunkai}@nankai.edu.cn

Yuan-An Xiao
Peking University
xiaoyuanan@pku.edu.cn

Siqi Ma
The University of New South Wales
siqi.ma@unsw.edu.au

Xiaofeng Chen
Xidian University
xfchen@xidian.edu.cn

The 32nd USENIX Security Symposium