NDSS 2026

南開大學 Nankai University

Targeted Password Guessing Using k-Nearest Neighbors

Zhen Li, Ding Wang*

College of Cryptology and Cyber Science, Nankai University, Tianjin, China

github:

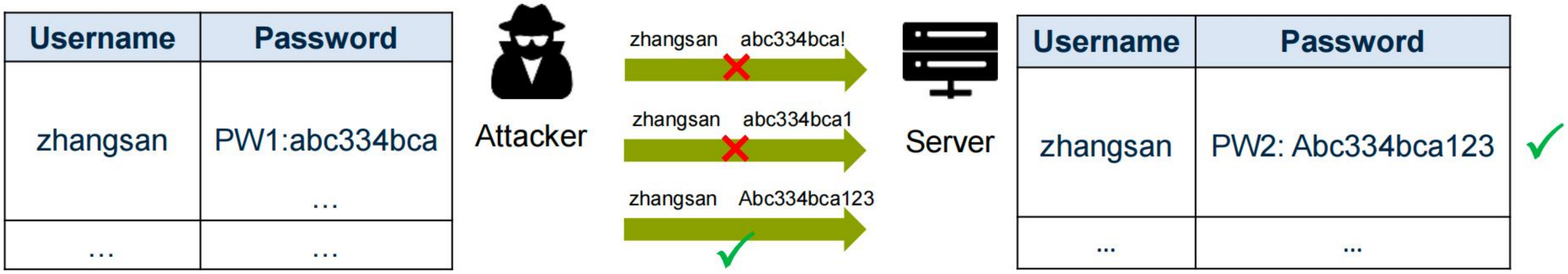
Passwords are irreplaceable

- **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

- **21%–33% of users tend to slightly modify their existing passwords when creating a new password.**
- **20%–59% of users directly reuse their existing passwords.**
- **Recent large-scale breaches, including CAM4 (10.8 billion credentials) and MOAB (26 billion credentials), have provided attackers with ample material for credential stuffing campaigns.**



Username	Password
zhangsan	PW1:abc334bca …
…	…

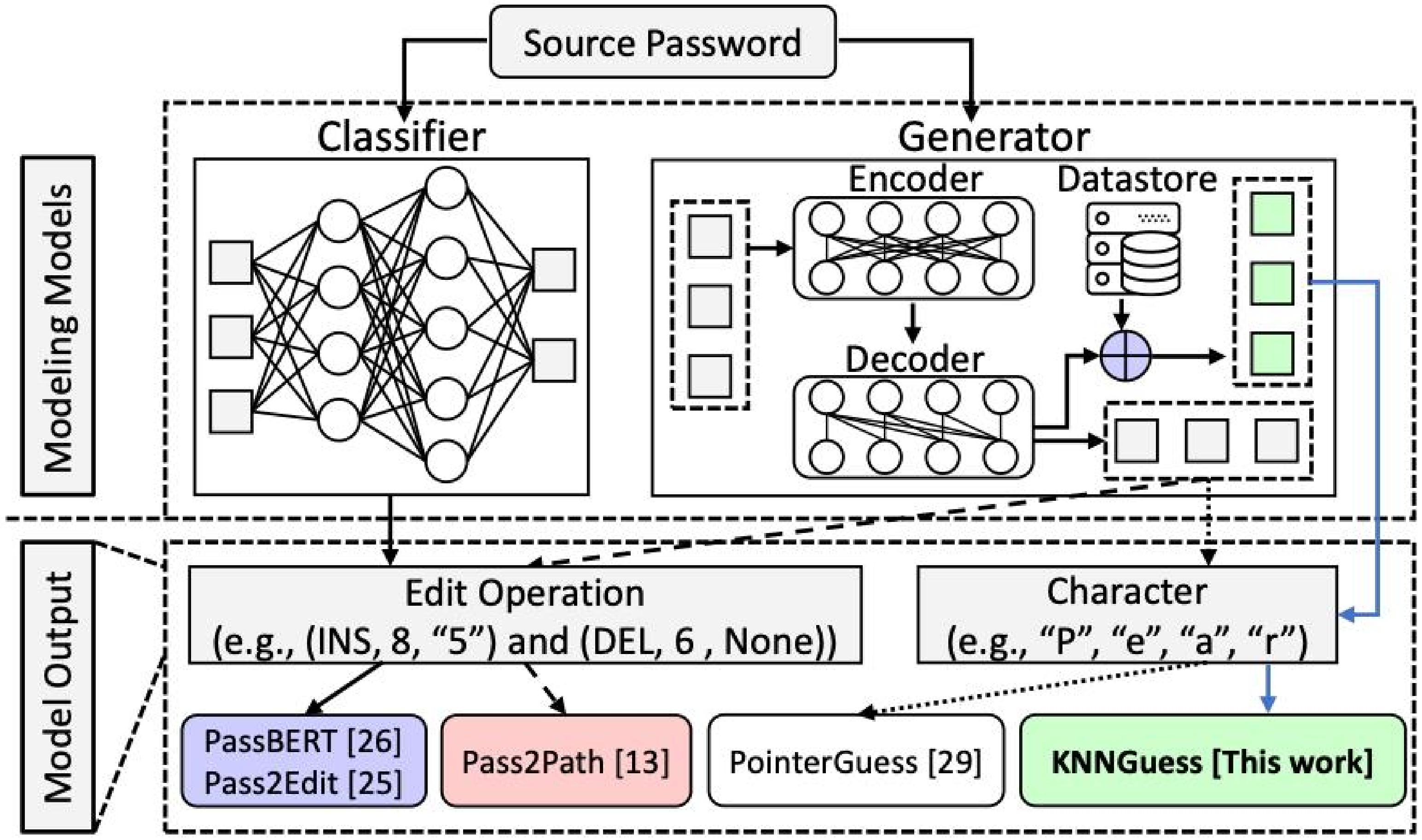
Username	Password
zhangsan	PW2: Abc334bca123
...	...

Three types of password reuse

- **Type–1: The user makes simple or moderate changes to the source password, and the new password is similar in structure to the source password (e.g., Shark0301 → shark03).**
- **Type–2: The user chooses to use a popular password (e.g., Shark0301 → loveu4ever). Such popular passwords often appear in publicly leaked dictionaries and are therefore insecure.**
- **Type–3: The user creates a new password based on a partial pattern of the source password. While the two passwords may appear "dissimilar", they could be intrinsically similar in semantics and vulnerable to attackers (e.g., Shark0301 → Bear11).**

Password probability modeling

- **Model input: user's old password character sequence PW1**



Capture Type-1 Password Reuse Behavior

- **Type-1: The user makes simple or moderate changes to the source password**
- **Model architecture：Transformer --> TransGuess**
- **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

Capture Type-2 Password Reuse Behavior

- **Type–2: The user chooses to use a popular password**
- **previous works：assumed that the probability of choosing a popular password is equal among all users.**

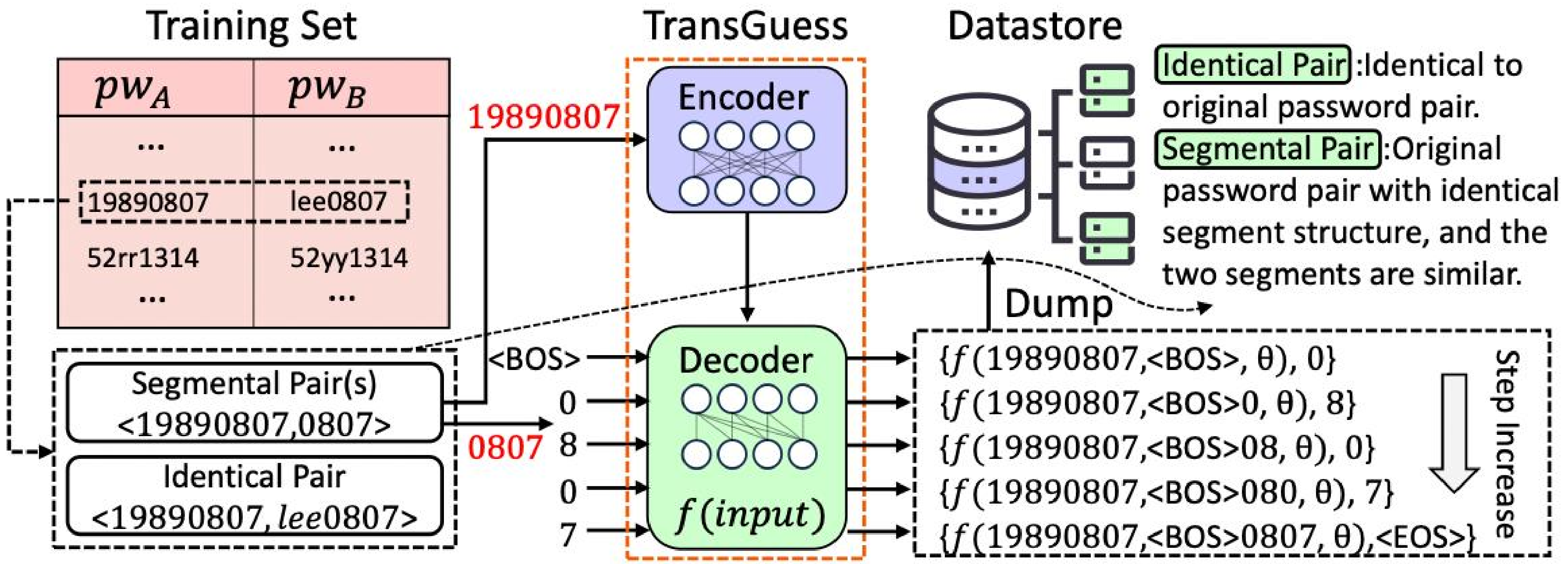
$$\boldsymbol{P_{popular}(pw_{target} \mid pw_{source}) = P_{popular}(pw_{target}) = List_p(pw_{target})}$$

- **our work: different source passwords (e.g., "p@ssw0rd" and "summer0803") may yield varying probabilities of generating the same popular password like "Passw0rd"**

$$\boldsymbol{P_{popular}(pw_{target} \mid pw_{source}) = Distance_h(pw_{target}, pw_{source}) * List_p(pw_{target})}$$

Capture Type-3 Password Reuse Behavior

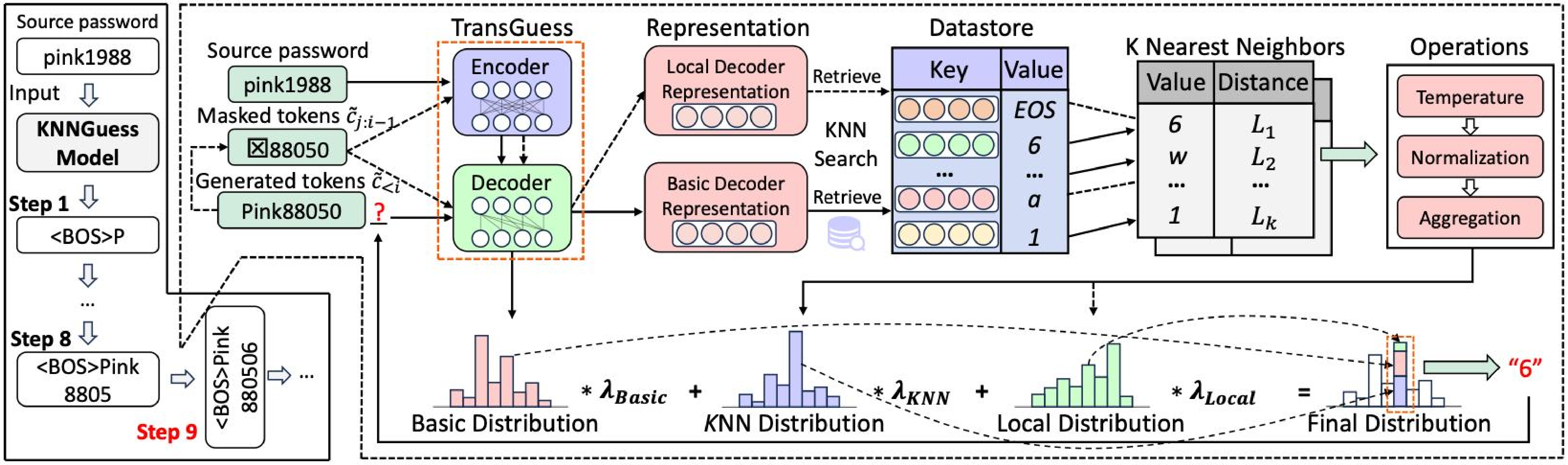
- **Type-3: The user creates a new password based on a partial pattern of the source password.**
- **KNN-TPG: 1) Build datastore: It learns to map a source password and the prefix of a target password to the next correct character in the latent space.**


Training Set
pw_A
pw_B
...
19890807
lee0807
52rr1314
52yy1314
Segmental Pair(s)
<19890807,0807>
Identical Pair
<19890807,lee0807>
19890807
0807
<BOS>
0
8
0
7
TransGuess
Encoder
Decoder
$f(input)$
Datastore
Identical Pair :Identical to original password pair.
Segmental Pair :Original password pair with identical segment structure, and the two segments are similar.
Dump
{f(19890807,<BOS>, θ), 0}
{f(19890807,<BOS>0, θ), 8}
{f(19890807,<BOS>08, θ), 0}
{f(19890807,<BOS>080, θ), 7}
{f(19890807,<BOS>0807, θ),<EOS>}
Step Increase

Capture Type-3 Password Reuse Behavior

- **KNN–TPG: 2) Generating passwords:**

$$\begin{aligned} P(\widetilde{c}_i|pw_A,\widetilde{c}_{<i}) &= P_{Basic}(\widetilde{c}_i|pw_A,\widetilde{c}_{<i},\theta)\cdot\lambda_{Basic} \\ &+ P_{KNN}(\widetilde{c}_i|pw_A,\widetilde{c}_{<i},d)\cdot\lambda_{KNN} \\ &+ P_{Local}(\widetilde{c}_i|\widetilde{c}_{j:i-1},d)\cdot\lambda_{Local}. \end{aligned}$$



Password datasets we used

- **From various high-profile websites**
- **Five Chinese datasets, five English datasets, and two mixed datasets**
- **Leaked between 2012-2021**
- **A total of 4.8 billion passwords**

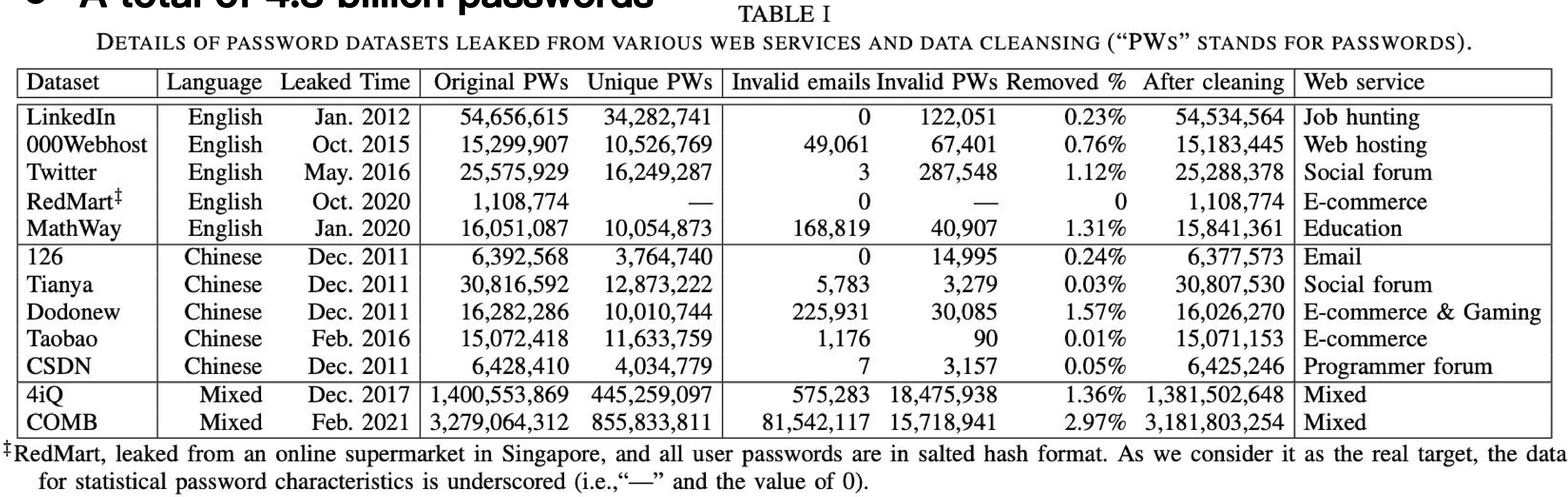


TABLE I

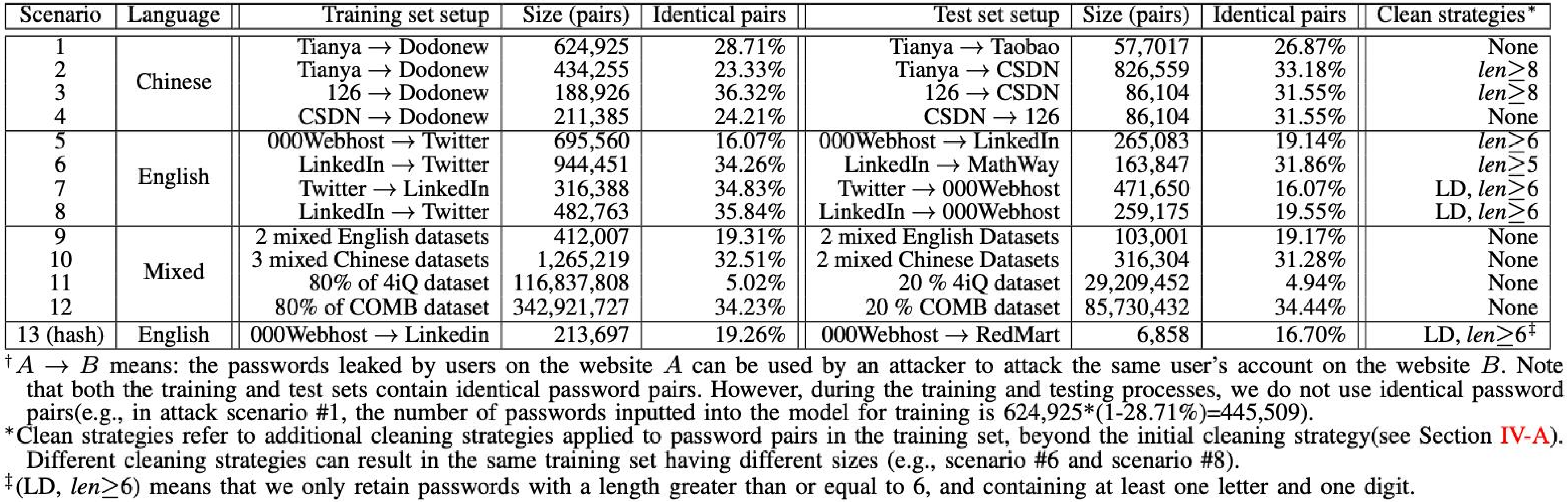
DETAILS OF PASSWORD DATASETS LEAKED FROM VARIOUS WEB SERVICES AND DATA CLEANSING ("PWs" STANDS FOR PASSWORDS).

Dataset	Language	Leaked Time	Original PWs	Unique PWs	Invalid emails	Invalid PWs	Removed %	After cleaning	Web service
LinkedIn	English	Jan. 2012	54,656,615	34,282,741	0	122,051	0.23%	54,534,564	Job hunting
000Webhost	English	Oct. 2015	15,299,907	10,526,769	49,061	67,401	0.76%	15,183,445	Web hosting
Twitter	English	May. 2016	25,575,929	16,249,287	3	287,548	1.12%	25,288,378	Social forum
RedMart‡	English	Oct. 2020	1,108,774	—	0	—	0	1,108,774	E-commerce
MathWay	English	Jan. 2020	16,051,087	10,054,873	168,819	40,907	1.31%	15,841,361	Education
126	Chinese	Dec. 2011	6,392,568	3,764,740	0	14,995	0.24%	6,377,573	Email
Tianya	Chinese	Dec. 2011	30,816,592	12,873,222	5,783	3,279	0.03%	30,807,530	Social forum
Dodonew	Chinese	Dec. 2011	16,282,286	10,010,744	225,931	30,085	1.57%	16,026,270	E-commerce & Gaming
Taobao	Chinese	Feb. 2016	15,072,418	11,633,759	1,176	90	0.01%	15,071,153	E-commerce
CSDN	Chinese	Dec. 2011	6,428,410	4,034,779	7	3,157	0.05%	6,425,246	Programmer forum
4iQ	Mixed	Dec. 2017	1,400,553,869	445,259,097	575,283	18,475,938	1.36%	1,381,502,648	Mixed
COMB	Mixed	Feb. 2021	3,279,064,312	855,833,811	81,542,117	15,718,941	2.97%	3,181,803,254	Mixed

‡RedMart, leaked from an online supermarket in Singapore, and all user passwords are in salted hash format. As we consider it as the real target, the data for statistical password characteristics is underscored (i.e.,"—" and the value of 0).

Experimental setup

- **13 attack scenarios**
- **specific cleaning methods will be applied based on the distinct website**
- **Hardware Configuration：NVIDIA RTX 3090 GPU (including 24GB of VRAM)**



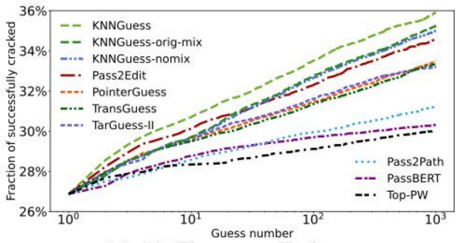
Scenario	Language	Training set setup	Size (pairs)	Identical pairs	Test set setup	Size (pairs)	Identical pairs	Clean strategies*
1	Chinese	Tianya $\rightarrow$ Dodonew	624,925	28.71%	Tianya $\rightarrow$ Taobao	57,7017	26.87%	None
2		Tianya $\rightarrow$ Dodonew	434,255	23.33%	Tianya $\rightarrow$ CSDN	826,559	33.18%	$len \geq 8$
3		126 $\rightarrow$ Dodonew	188,926	36.32%	126 $\rightarrow$ CSDN	86,104	31.55%	$len \geq 8$
4		CSDN $\rightarrow$ Dodonew	211,385	24.21%	CSDN $\rightarrow$ 126	86,104	31.55%	None
5	English	000Webhost $\rightarrow$ Twitter	695,560	16.07%	000Webhost $\rightarrow$ LinkedIn	265,083	19.14%	$len \geq 6$
6		LinkedIn $\rightarrow$ Twitter	944,451	34.26%	LinkedIn $\rightarrow$ MathWay	163,847	31.86%	$len \geq 5$
7		Twitter $\rightarrow$ LinkedIn	316,388	34.83%	Twitter $\rightarrow$ 000Webhost	471,650	16.07%	LD, $len \geq 6$
8		LinkedIn $\rightarrow$ Twitter	482,763	35.84%	LinkedIn $\rightarrow$ 000Webhost	259,175	19.55%	LD, $len \geq 6$
9	Mixed	2 mixed English datasets	412,007	19.31%	2 mixed English Datasets	103,001	19.17%	None
10		3 mixed Chinese datasets	1,265,219	32.51%	2 mixed Chinese Datasets	316,304	31.28%	None
11		80% of 4iQ dataset	116,837,808	5.02%	20 % 4iQ dataset	29,209,452	4.94%	None
12		80% of COMB dataset	342,921,727	34.23%	20 % COMB dataset	85,730,432	34.44%	None
13 (hash)	English	000Webhost $\rightarrow$ Linkedin	213,697	19.26%	000Webhost $\rightarrow$ RedMart	6,858	16.70%	LD, $len \geq 6$[‡]

[†] $A \rightarrow B$ means: the passwords leaked by users on the website A can be used by an attacker to attack the same user's account on the website B. Note that both the training and test sets contain identical password pairs. However, during the training and testing processes, we do not use identical password pairs(e.g., in attack scenario #1, the number of passwords inputted into the model for training is 624,925*(1-28.71%)=445,509).

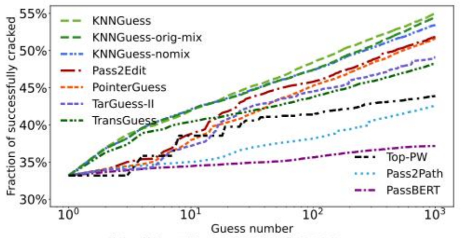
*Clean strategies refer to additional cleaning strategies applied to password pairs in the training set, beyond the initial cleaning strategy(see Section IV-A). Different cleaning strategies can result in the same training set having different sizes (e.g., scenario #6 and scenario #8).

[‡](LD, $len \geq 6$) means that we only retain passwords with a length greater than or equal to 6, and containing at least one letter and one digit.

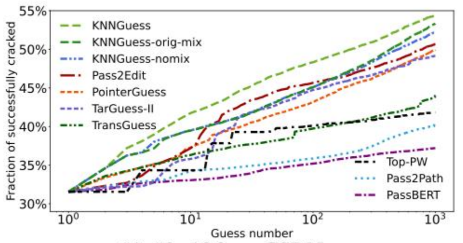
Experimental results



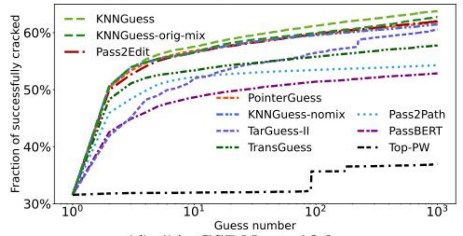
(a) #1: Tianya → Taobao



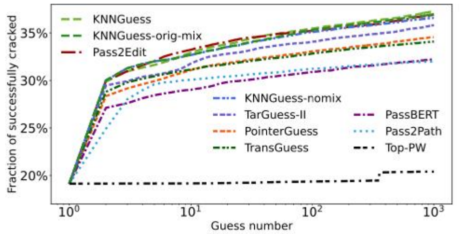
(b) #2: Tianya → CSDN



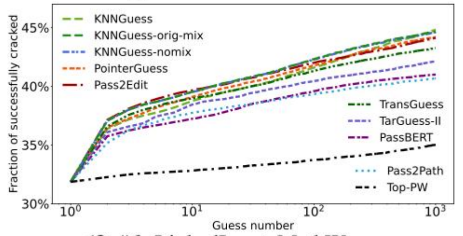
(c) #3: 126 → CSDN



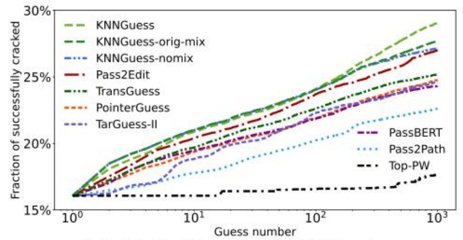
(d) #4: CSDN → 126



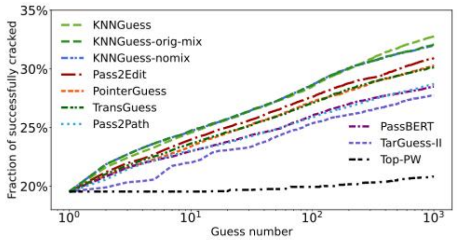
(e) #5: 000Webhost → LinkedIn



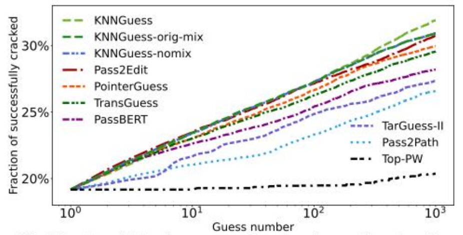
(f) #6: LinkedIn → MathWay



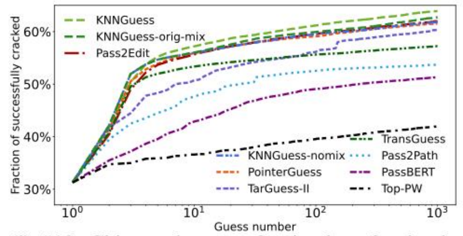
(g) #7: Twitter → 000Webhost



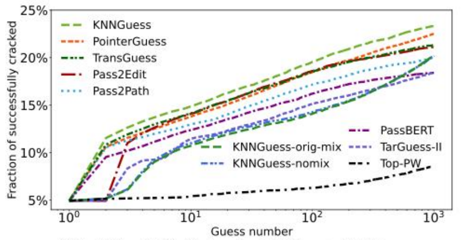
(h) #8: LinkedIn → 000Webhost



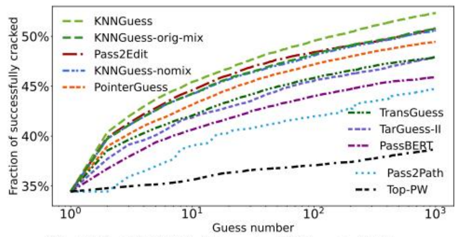
(i) #9: English datasets: 2 mixed → 2 mixed



(j) #10: Chinese datasets: 3 mixed → 2 mixed



(k) #11: 4iQ dataset: 80% → 20%



(l) #12: COMB dataset: 80% → 20%

- **Within 100 guesses, the guessing success rates of our KNNGuess are 8.52%–27.66% higher than state-of-the-art password guessing models.**

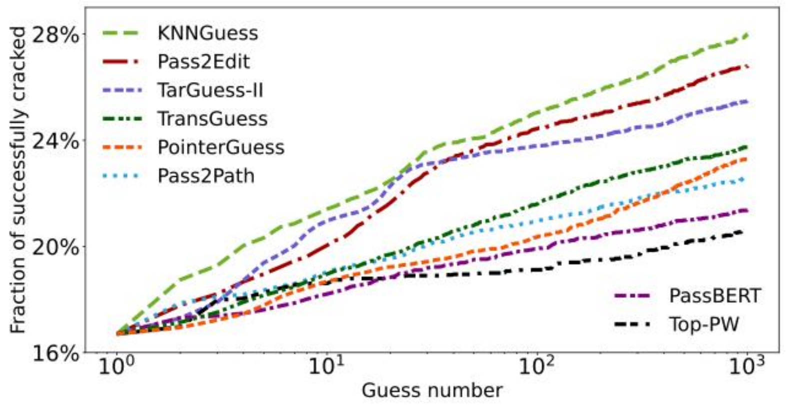
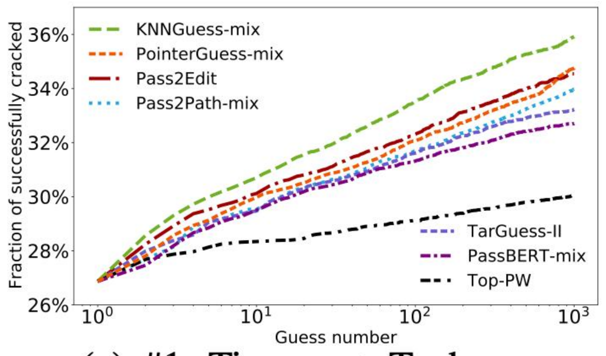


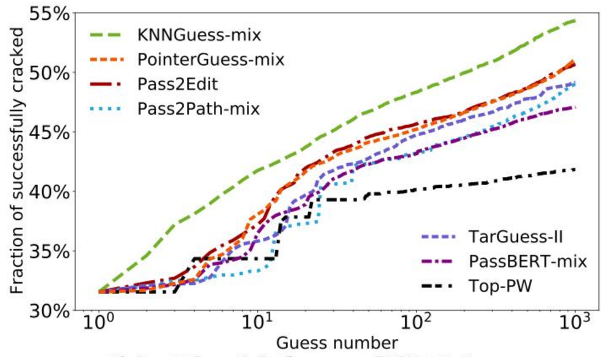
Fig. 6. The experimental results in the salted hash attack scenario #13. The training and test sets are listed in Table II. Our KNNGuess model still achieves the highest cracking success rate compare with other models.

New mixing popular password method

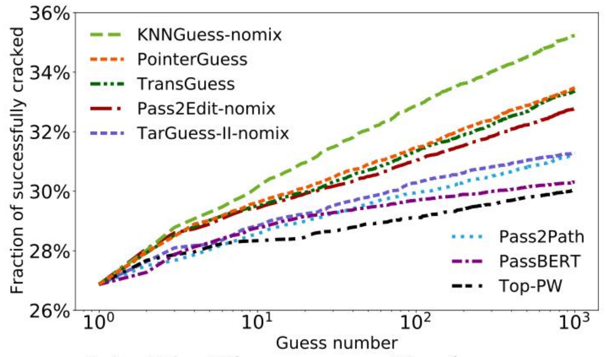
- **Our new mixing popular password method improves the guessing success rate by 9.21% compared to the mixing popular password method used in previous work.**



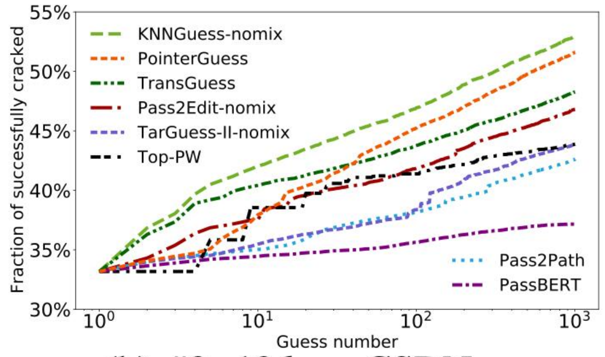
(a) #1: Tianya → Taobao



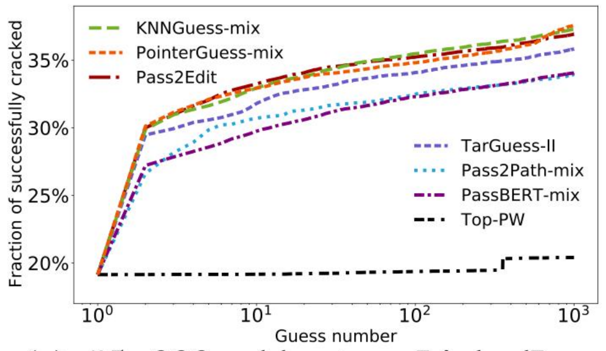
(b) #3: 126 → CSDN



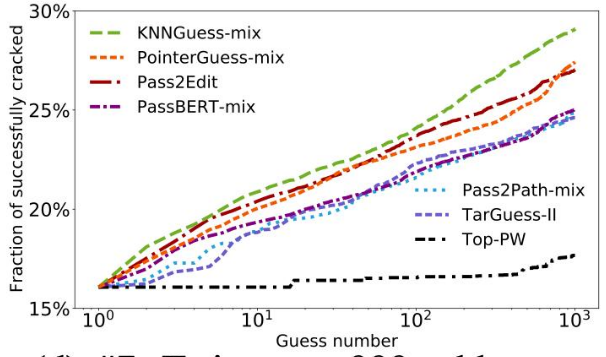
(a) #1: Tianya → Taobao



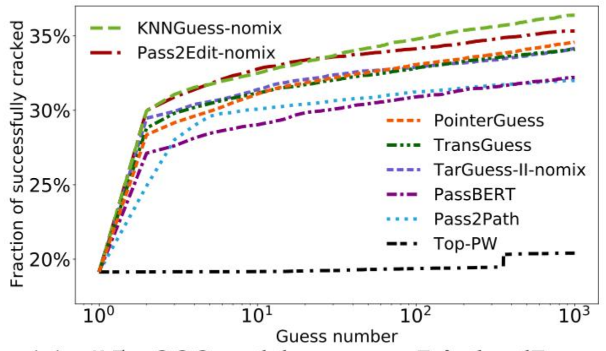
(b) #3: 126 → CSDN



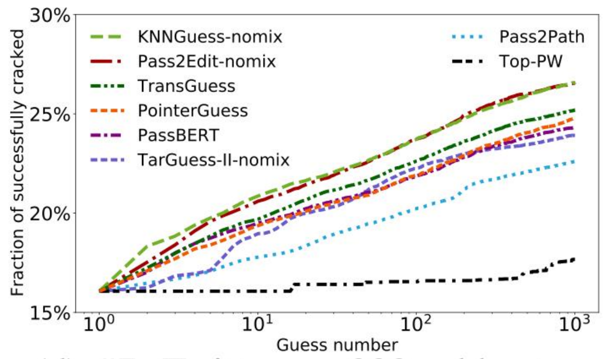
(c) #5: 000webhost → LinkedIn



(d) #7: Twitter → 000webhost



(c) #5: 000webhost → LinkedIn



(d) #7: Twitter → 000webhost

Analysis of cracked passwords

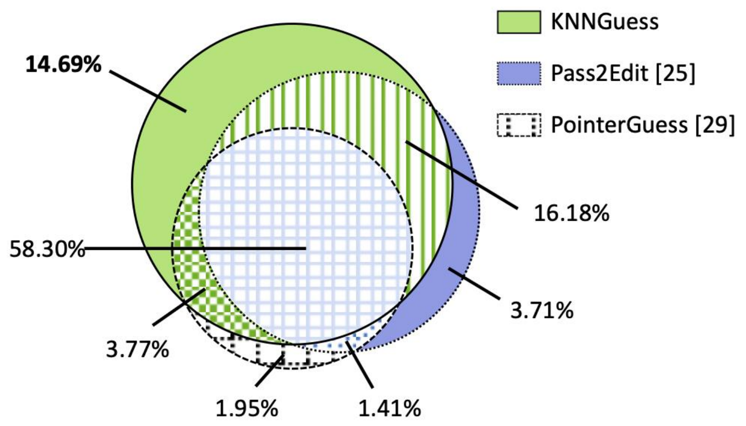
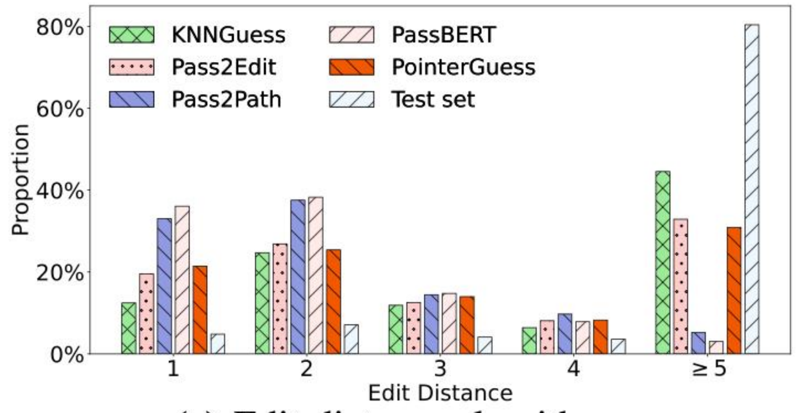


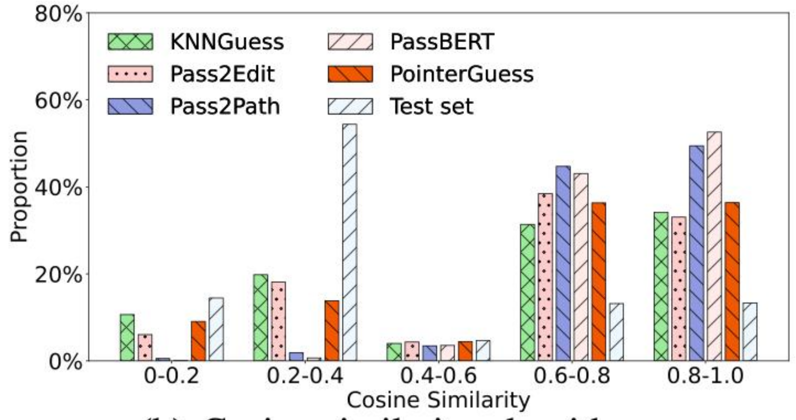
Fig. 9. The overlap ratio of passwords cracked by KNNGuess, Pass2Edit and PointerGuess. Our KNNGuess model has the highest number of cracked password pairs and the highest individual cracking ratio (i.e., 14.69%).

Model name	Type-1	Type-2	Type-3
KNNGuess (This work)	27994 (79.98%)	6370 (18.2%)	636 (1.82%)
Pass2Edit [25]	26025 (79.85%)	6284 (19.28%)	283 (0.87%)
PointerGuess [29]	21884 (84.31%)	3719 (14.33%)	355 (1.4%)
Pass2Path [13]	15368 (99.11%)	7 (0.05%)	131 (0.84%)
PassBERT [26]	19242 (99.73%)	3 (0.02%)	49 (0.25%)
TarGuess-II [28]	21394 (76.95%)	6156 (22.14%)	254 (0.91%)

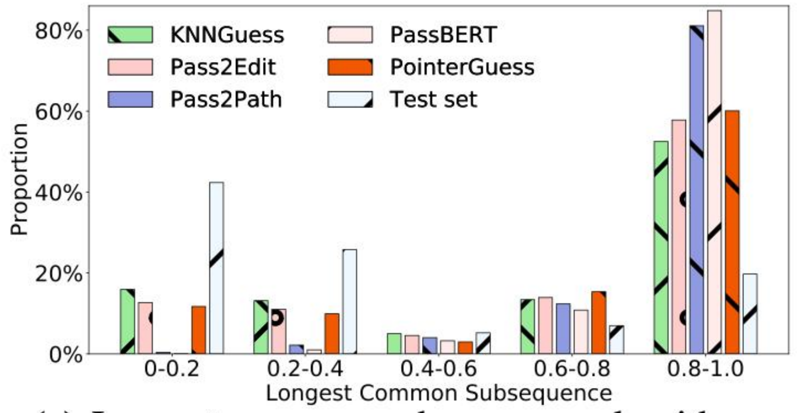
†We defined three types of user reuse behaviors in Sec. I-A. For example, Type-1 means that user makes simple changes to the source password.



(a) Edit distance algorithm



(b) Cosine similarity algorithm



(c) Longest common subsequence algorithm

Fig. 10. The similarity distribution of password pairs cracked by 5 attack models. Figs. 10(a)-10(c) show the results using spatial distance metrics and sequence alignment-based metrics (i.e., edit distance, cosine similarity, and longest common subsequence (LCS) algorithm), to measure the similarity distribution of cracked passwords. The "Test set" represents all password pairs in the test set. KNNGuess is particularly good at predicting distant password reuse behaviors.

NDSS 2026

Nankai University

Thank you!

Targeted Password Guessing Using k-Nearest Neighbors

Zhen Li, Ding Wang *

College of Cryptology and Cyber Science, Nankai University, Tianjin, China

github: