PointerGuess: Targeted Password Guessing Model using Pointer Mechanism

Kedong Xiu

Nankai University

kedongxiu@mail.nankai.edu.cn

Ding Wang

Nankai University

wangding@nankai.edu.cn

Textual Passwords

- Easy to use
- Low cost
- Easy to change

Password still remains its dominance in the future

Credential Stuffing Attack: A realistic threat for online users

- Web users have **80-107 (avg.)** passwords[2].
- **58%~79%** users directly reuse or slightly modify their existing passwords [3-6].
- Latest DBIR reports that **77%** web attack is credential stuffing attack [1].

Username	Password
Jack	IloveMacOP
...	...

Leaked Dataset

Attacker

Username	Password
Jack	MacOP5789
...	...

Target Server

[1] https://www.verizon.com/business/de-de/resources/reports/2024/dbir/2024-dbir-data-breach-investigations-report.pdf
[2] https://www.lastpass.com/resources/ebook/psychology-of-passwords-2021
[3] https://www.zdnet.com/article/google-launches-password-checkup-feature-will-add-it-to-chrome-later-this-year/
[4] https://services.google.com/fh/files/blogs/google_security_infographic.pdf
[5] Beyond Credential Stuffing: Password Similarity Models Using Neural Networks
[6] fuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars

Prior work

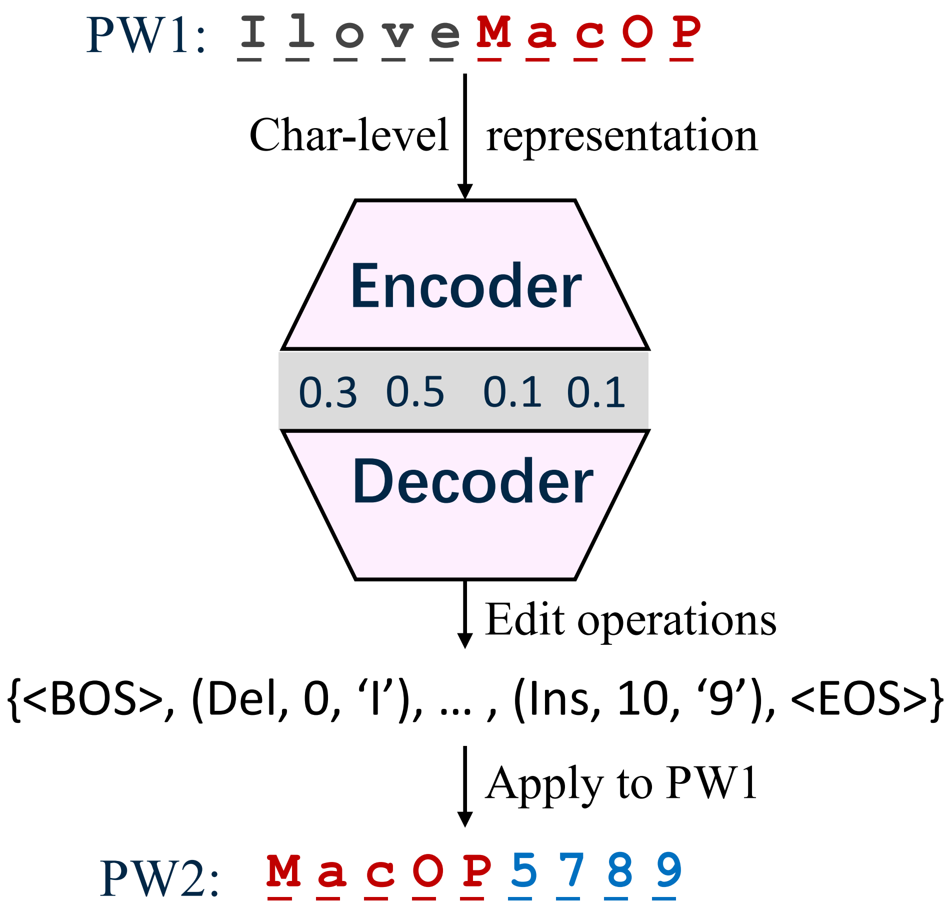
“Password-to-Path”-based models

Pass2Edit

PassBERT

Pass2Path

Common idea:
Conditional password guessing is a process of predicting edit operations based the old password.



Existing issues

1. **Existing models need to filter training set while overlooking similar password pairs**

User	PW1	PW2	Edit distance	Cosine similarity
A	3080124	cooper3080124	4 ✓	0.71 ✓
B	720710	720710720710	6 ✗	0.95 ✓
C	IloveMacOP	MACOP	7 ✗	0.25 ✗
D	iloveu4ever	ILOVEU4EVER	10 ✗	0 ✗

- Pass2Path uses **edit distance >= 4** to filter training set
- Pass2Edit users **cosine similarity > 0.3** to filter training set

Existing issues

1. **Existing models need to filter training set while overlooking similar password pairs**
2. **Heuristic method to mix popular passwords**

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
…	…	…

Beam Search Result

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

Mixed Password List

Prob (exponential)	Password
-3.14	12345678
-3.28	123456789
-4.41	11111111
-4.93	dearbook
…	…

Popular Passwords

Figure 4 in [1]

[1] Ding Wang, Yunkai Zou, Yuan-an Xiao, Siqi Ma and Xiaofeng Chen, "Pass2Edit: A Multi-Step Generative Model for Guessing Edited Passwords", in Proc. USENIX SEC 2023

Existing issues

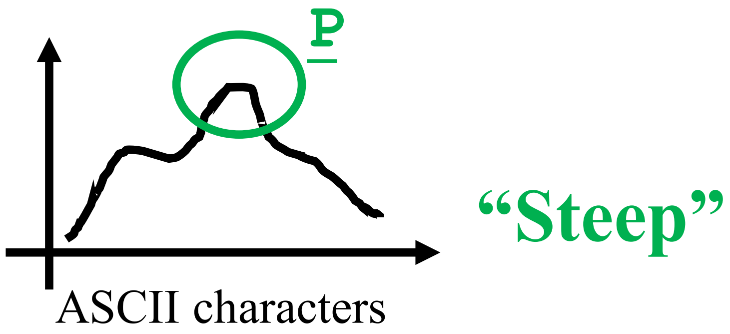
1. Existing models need to filter training set while overlooking similar password pairs
2. Heuristic method to mix popular passwords
3. The large number of atomic edit operations

Encoder input: I l o v e M a c O P

Decoder input: M a c O ⊠

Target password: M a c O P 5 7 8 9

predict

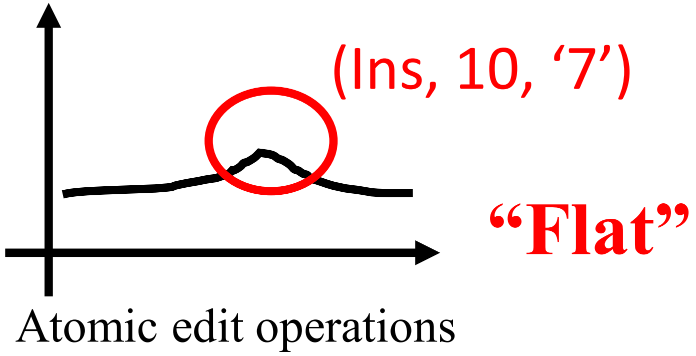


Encoder input: I l o v e M a c O P

Decoder input: {<BOS>, …, (Ins, 10, '5')}

Target password: M a c O P 5 7 8 9

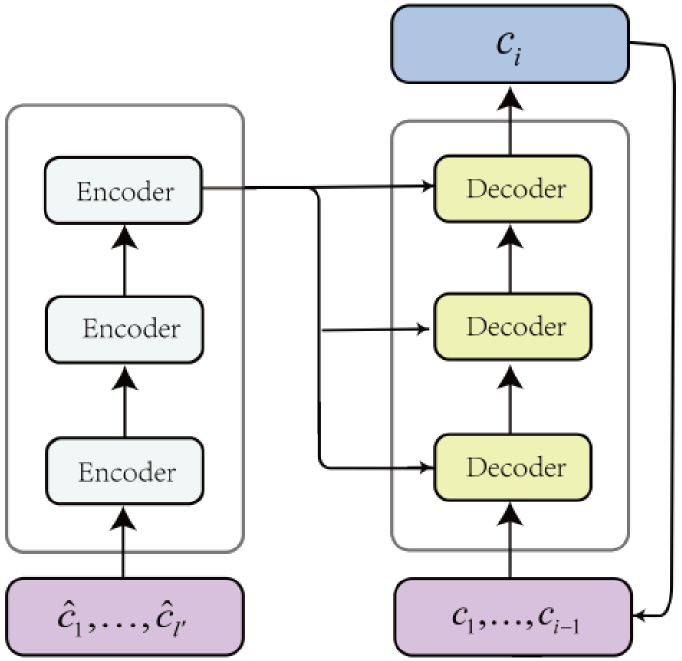
predict



Existing issues

1. **Existing models need to filter training set while overlooking similar password pairs**
2. **Heuristic method to mix popular passwords**
3. **The large number of atomic edit operations**
4. **The inefficient utilization of the old password**

- Only **generate** "new" characters based on the model
- **Overlook** the copy operation from the old password



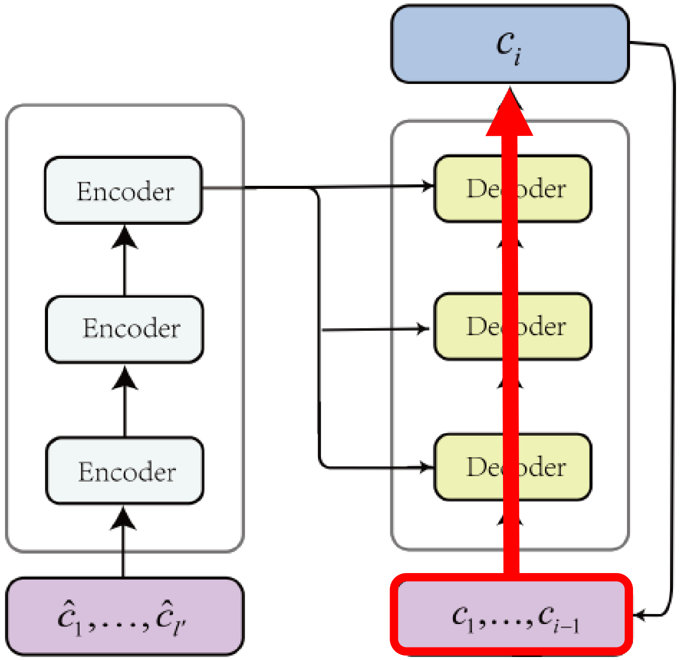
Passtrans model architecture[1]

[1] Xiaoxi He, Haibo Cheng, Jiahong Xie, Ping Wang, Kaitai Liang, "Passtrans: An Improved Password Reuse Model Based on Transformer", in Proc. ICASSP 2022.

Existing issues

1. Existing models need to filter training set while overlooking similar password pairs
2. Heuristic method to mix popular passwords
3. The large number of atomic edit operations
4. The inefficient utilization of the old password

- Only **generate** "new" characters based on the model
- **Overlook** the copy operation from the old password



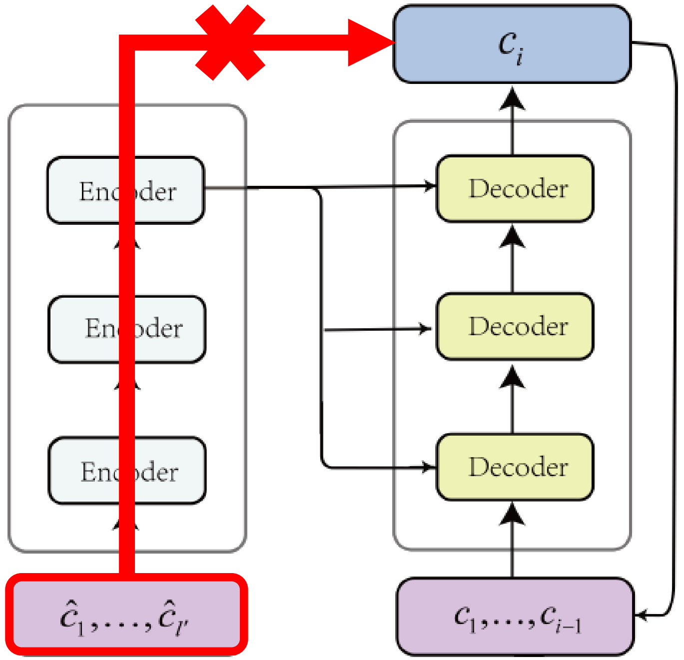
Passtrans model architecture[1]

[1] Xiaoxi He, Haibo Cheng, Jiahong Xie, Ping Wang, Kaitai Liang, "Passtrans: An Improved Password Reuse Model Based on Transformer", in Proc. ICASSP 2022.

Existing issues

1. **Existing models need to filter training set while overlooking similar password pairs**
2. **Heuristic method to mix popular passwords**
3. **The large number of atomic edit operations**
4. **The inefficient utilization of the old password**

- Only **generate** "new" characters based on the model
- **Overlook** the copy operation from the old password



Passtrans model architecture[1]

[1] Xiaoxi He, Haibo Cheng, Jiahong Xie, Ping Wang, Kaitai Liang, "Passtrans: An Improved Password Reuse Model Based on Transformer", in Proc. ICASSP 2022.

Existing issues

1. Existing models need to filter training set while overlooking similar password pairs
2. Heuristic method to mix popular passwords
3. The large number of atomic edit operations
4. The inefficient utilization of the old password

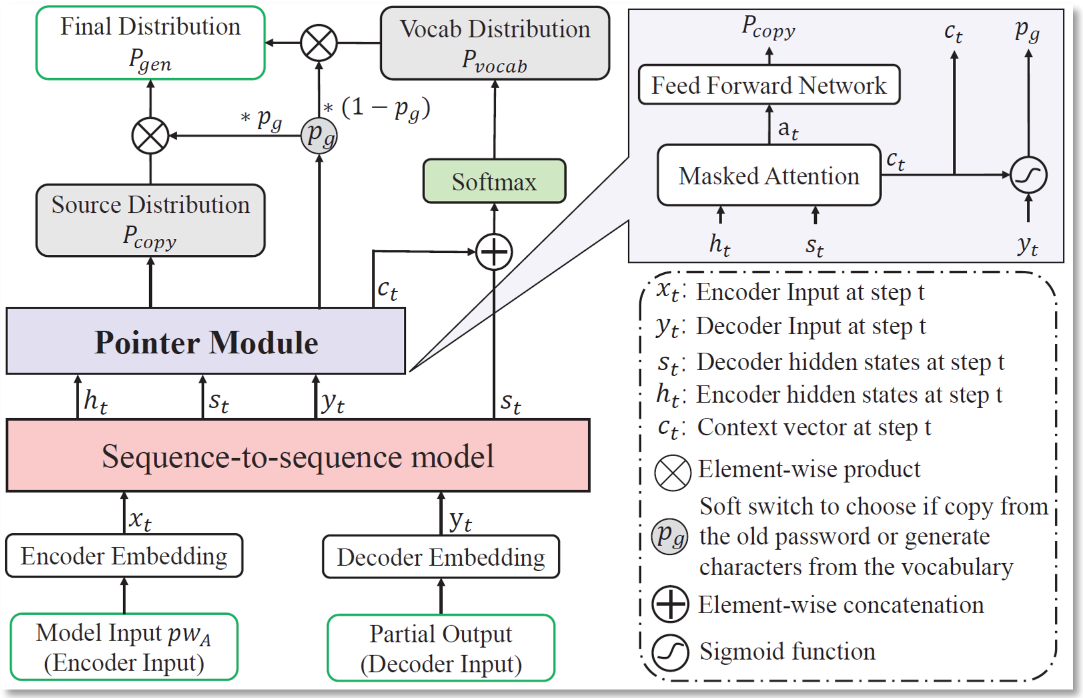
Our work

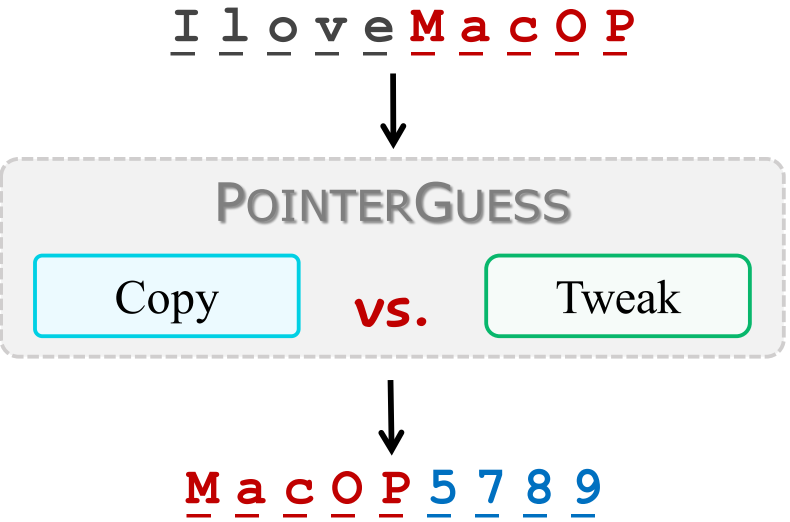
- ☐ **Directly predict character sequence of the target password**
- ☐ **Model a new conditional password guessing probability**
- ☐ **Consider both copying characters from the old password and generating new characters**

PointerGuess: Targeted Password Guessing Model

Modeling new conditional password guessing probability

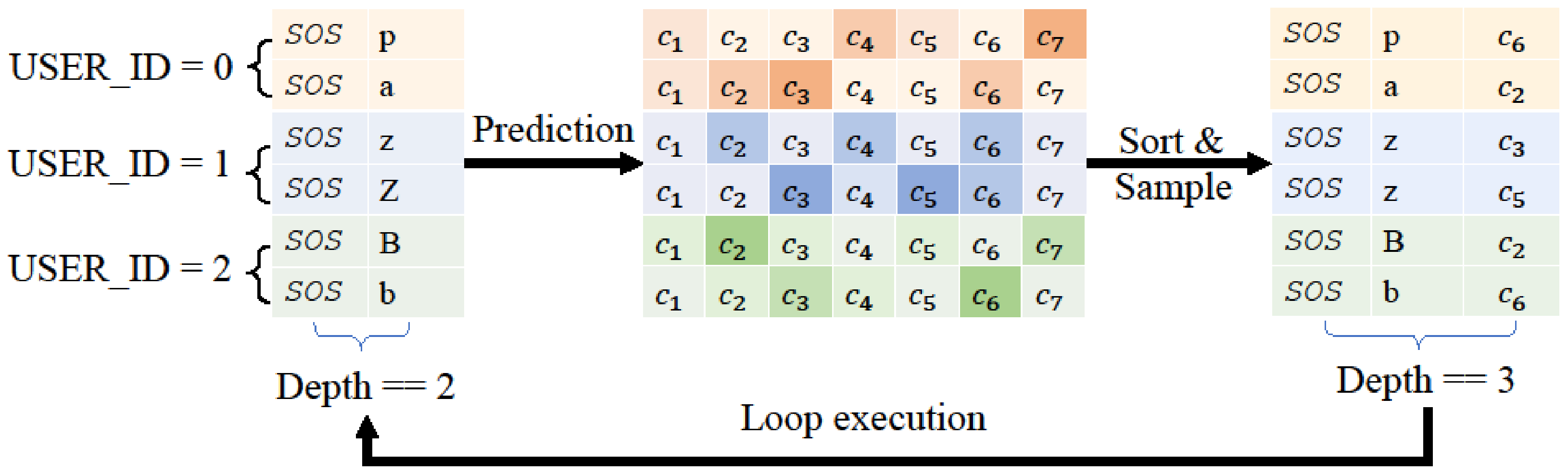
- Directly **copy** characters from pw_A, i.e., $P_{copy} = FFN\left(\sum_{\{j:c_j=c\}} \boldsymbol{a_j^i}\right)$
- **Generate** characters based on pw_A, i.e., $P_{vocab} = softmax(W' * (W * [s_t, c_t] + b_{out}) + b'_{out})$
- **Weighted-sum** two conditional probabilities, i.e., $P_{gen} = \boldsymbol{p_g} * P_{copy} + (1 - \boldsymbol{p_g}) * P_{vocab}$





Facilitate password generation

- Implement Batch beam search algorithm for password generation
  - Choose the **batch size** before generating guesses.
  - Set the **global topK guesses** for each user (e.g., 1000 guesses every user).
  - Set the **local topK candidates** for every generation (e.g., 7 candidates).



3~4 times faster

Experimental setup

- Attack scenario construction
  - 11 real-world datasets (4 Chinese datasets, 5 English datasets and 2 large-scale mixed datasets)
  - 4 attack scenarios for Chinese and English, respectively
  - 4 large-scale attack scenarios
- Experiment environment
  - Running on NVIDIA RTX 3090 (24 GB of vRAM)

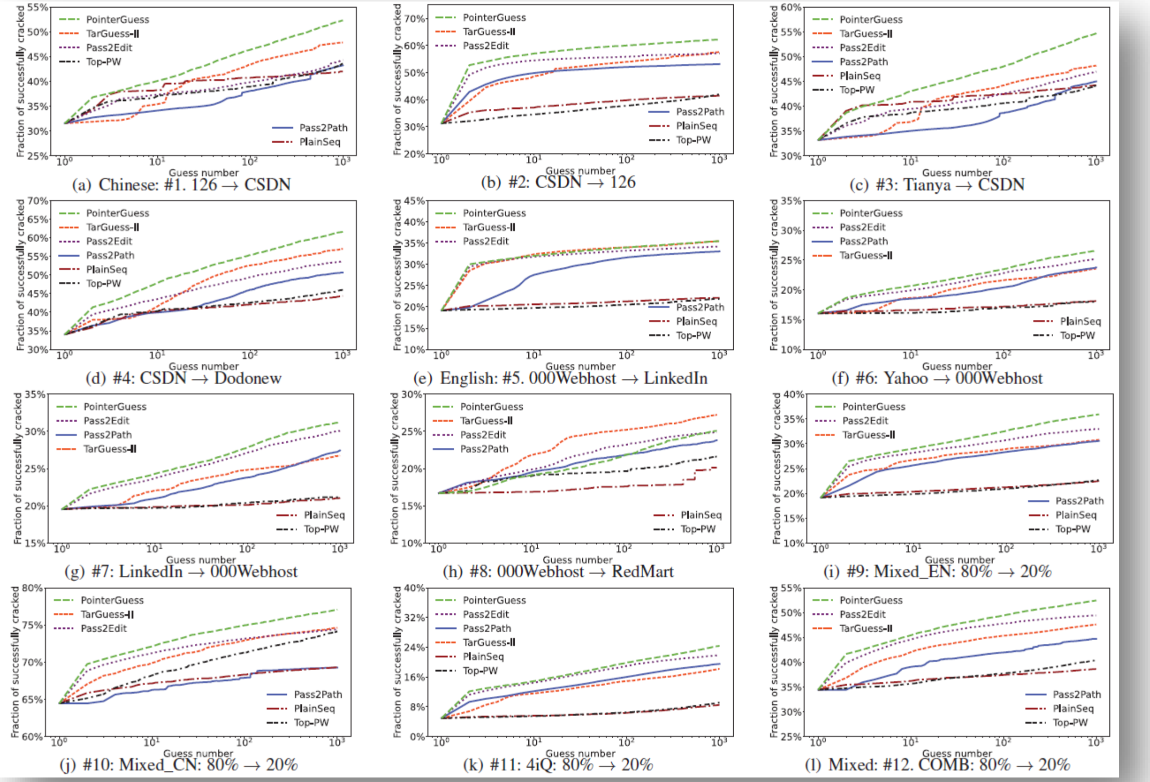
Everyone can have it!!

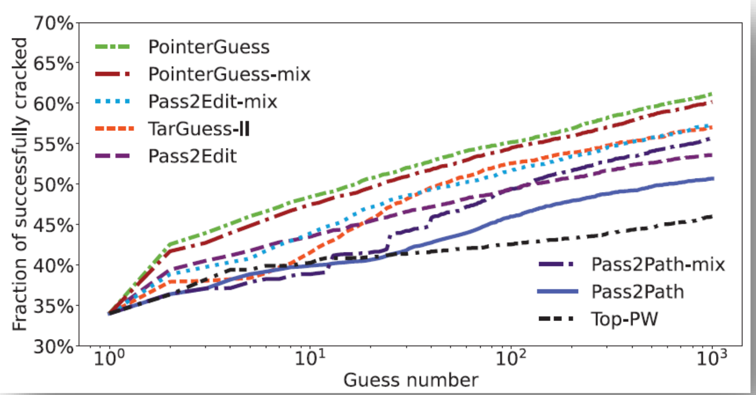
  - Randomly select 20,000 password pairs as test set

#. Attack scenario	Language	Training set setup	Size (pairs)	Testing set setup	Size (pairs)	Clean strategies†
#1. 126 → CSDN	Chinese	126 → Dodonew	188,926	126 → CSDN	85,206	Len≥8
#2. CSDN → 126	Chinese	CSDN → Dodonew	211,385	CSDN → 126	86,104	*Basic*
#3. Tianya → CSDN	Chinese	Tianya → Dodonew	434,255	Tianya → CSDN	826,559	Len≥8
#4. CSDN → Dodonew	Chinese	CSDN → 126	86,104	CSDN → Dodonew	211,385	*Basic*
#5. 000Webhost → LinkedIn	English	000Webhost → Yahoo	265,083	000Webhost → LinkedIn	213,697	Len≥6
#6. Yahoo → 000Webhost	English	Yahoo → LinkedIn	40,646	Yahoo → 000Webhost	37,479	LD
#7. LinkedIn → 000Webhost	English	LinkedIn → Yahoo	40,812	LinkedIn → 000Webhost	259,175	LD, Len≥6
#8. 000Webhost → RedMart	English	000Webhost → Linkedin	213,697	000Webhost → RedMart	6,858	Len≥6
#9. 80% Mixed_EN → 20% Mixed_EN	English	80% of Mixed_EN	338,857	20% of Mixed_EN	84,714	*Basic*
#10. 80% Mixed_CN → 20% Mixed_CN	Chinese	80% of Mixed_CN	434,255	20% of Mixed_CN	108,564	*Basic*
#11. 80% 4iQ → 20% 4iQ	Mixed	80% of 4iQ dataset	116,837,808	20 % 4iQ dataset	29,209,452	*Basic*
#12. 80% COMB → 20% COMB	Mixed	80% of COMB	342,921,727	20 % COMB dataset	85,730,432	*Basic*

Experimental results

- Within 100 guesses, the average success rate of POINTERGUESS is **21.23%~71.54%** (38.37% on average) higher than its foremost counterparts.
- POINTERGUESS inherently owns the ability of **generating popular passwords**.
- POINTERGUESS is **3~4 times faster** than other models while generating guesses.



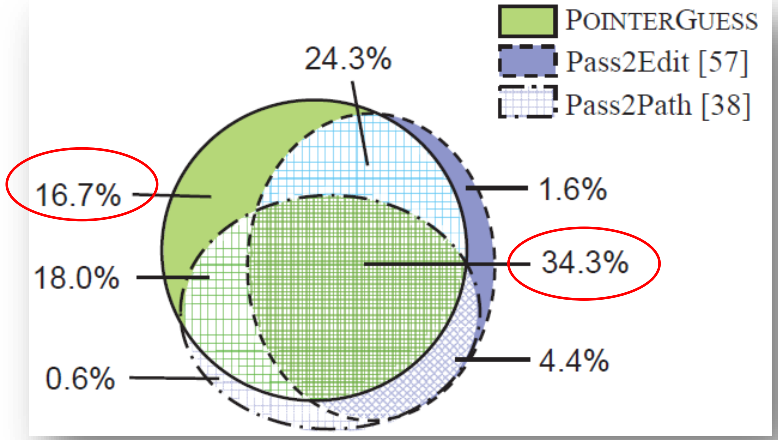


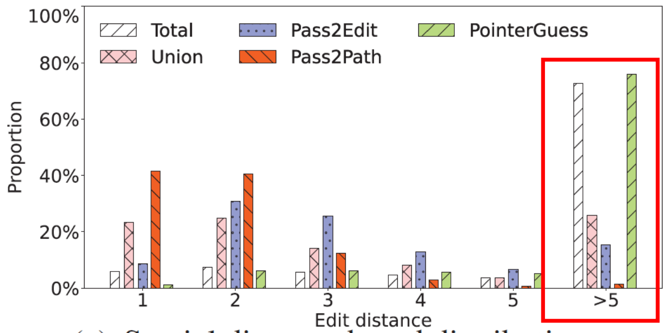
Attack model	POINTERGUESS	Pass2Edit [57]	Pass2Path [38]
Training time	15:14	**09:43**	14:10
Testing time	**00:24**	02:26	01:47
Speed‡ (pw/s)	**9,700~9,800**	2,100~2,200	2,900~3,000
Model size (MB)	**2.26**	11	53.6

Experiment analysis

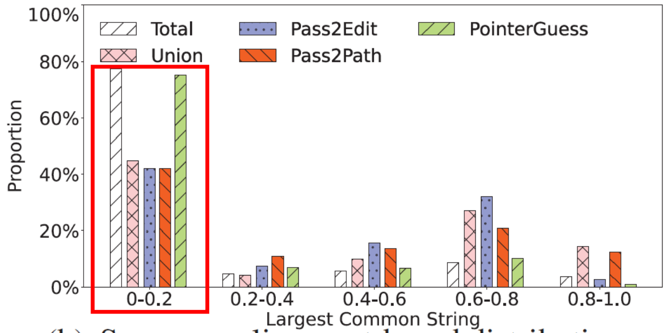
Overall analysis

Models	PointerGuess		Pass2Edit [57]		Pass2Path [38]	
Index	Old password	Target password	Old password	Target password	Old password	Target password
1	852255685145294	abc123	MCfaraona020591	mcfaraona91	8841800lin	lin8841800lin
2	boy78697740	boy123456789	edwardcullenqwe	Edwardcullen	jangobango88	jangobango1988
3	kazeevatanyuffka872ghbrjyf	kazeevatanyuffka	Castor	Castor08	13197277038	131w97277038
4	katmarlzelda969	katmarlzelda969@yahoo.com	4.14495E	4.14495E+13	IloveYOU2998	iloveyou2998
5	ghostgamer-2001	ghostgamer-2001@hotmail.com	t0romerda.	toromerda	SAIIIOK	sailiok
6	uuDBUMDM5NApOzYW	qweasdzxc	UHJVuhjvbr49	Uhjvuhjvbr49	wgpfuqd861208	wgpfUQD861208
7	jaydiltddasilva@partners.org	jaydilla1	30061986123	30061986qwe	rajuraju	raju2raju
8	102457685&	102457685!!	WMOOLMAN1058	WMOOLMAN	drdeath	1DRDEATH
9	1991322322	1.99132E+12	RBV//1960	rbl//1960	samantha	s@mantha
10	6125251987110	6.12525E+12	SharmaHellV1.0	HellV1.0	liljojo202	liljojo120





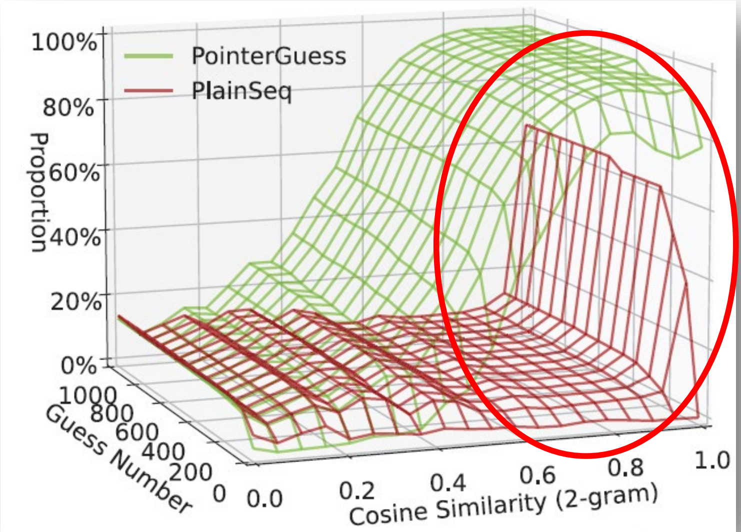
(a) Spatial distance-based distribution

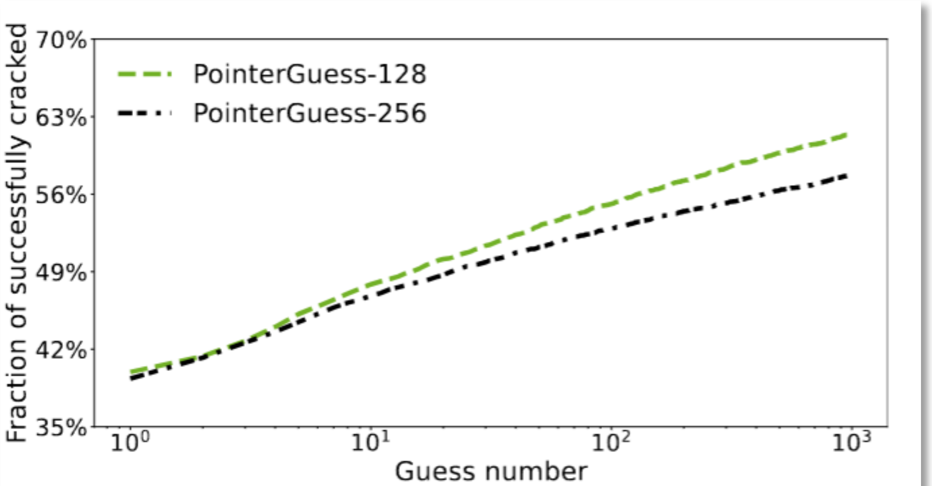


(b) Sequence alignment-based distribution

Experiment analysis

- Ablation study





(a) #4: CSDN → Dodonew

- Create **unique** passwords
  - E.g., 585129wupan → 585129
- Larger **similarity differences** between password pairs

- Model dimension **barely impacts** the model performance

Question:

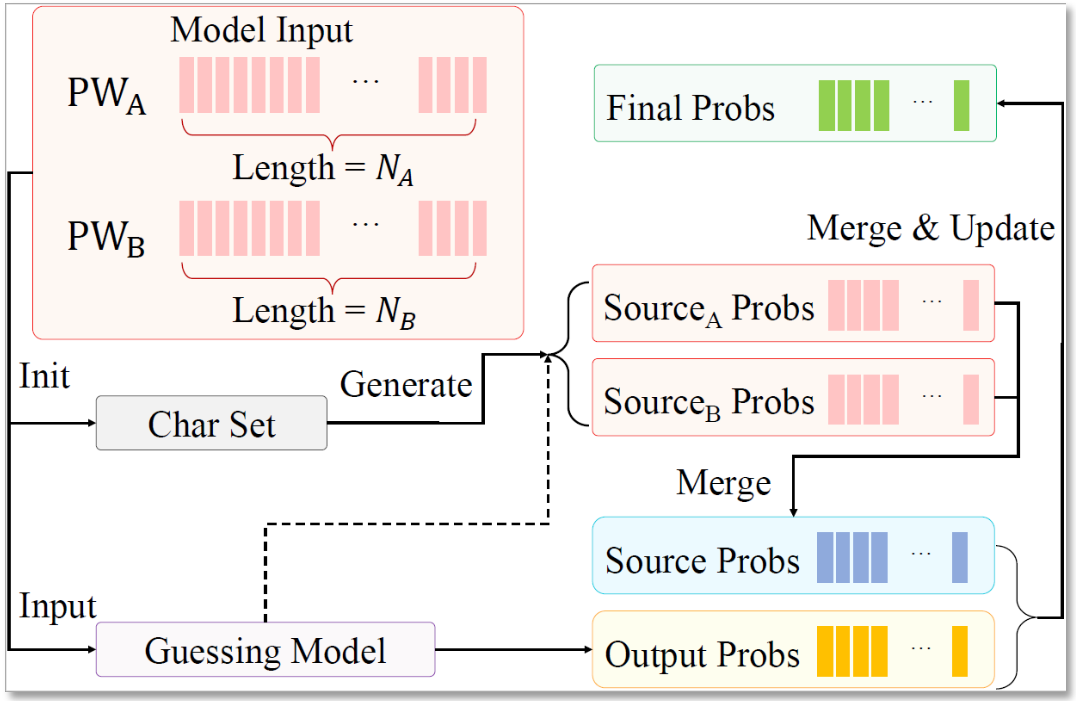
Can we use each victim’s multiple leaked passwords?

Answer:

YES! Why not?

Extensive work

MS-PointerGuess: Password Guessing Model based on Multi-Encoder Module



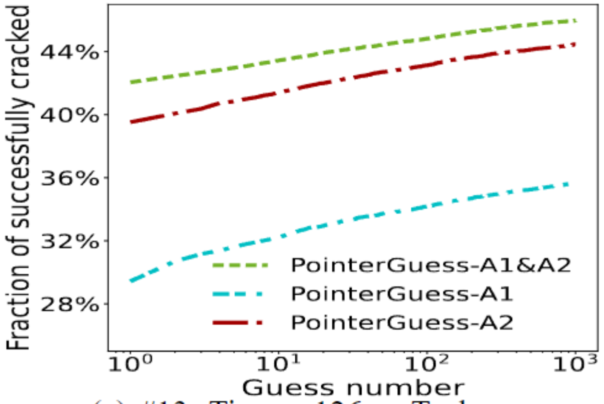
- Employ the pointer mechanism.
- Multiple encoders **parallel** process multiple old passwords for each user.
- Different encoders are assigned weight vectors that **sum up to 1**.

Extensive work

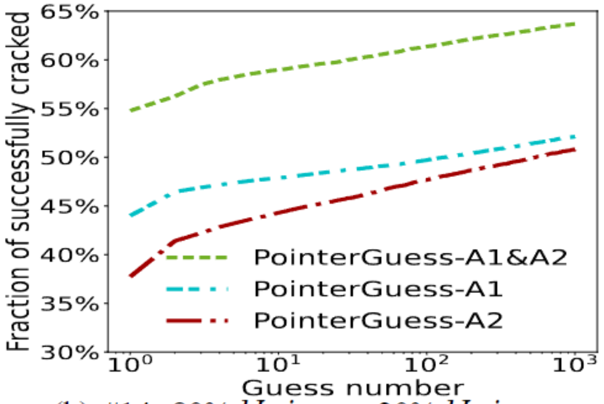
- Experiment setup
  - Six datasets (Tianya, 126, Taobao, Clixsense, LiveAuctioneers and 4iQ)
  - Two attack scenarios #13 and #14

#13A. Tianya, 126 → Taobao #13B. Tianya → Taobao #13C. 126 → Taobao	Chinese	Tianya, 126 → Dodonew Tianya → Dodonew 126 → Dodonew	95,457	Tianya, 126 → Taobao Tianya → Taobao 126 → Taobao	79,562	*Basic* *Basic* *Basic*
#14A. 80% *Union* → 20% $Union_B$* #14B. 80% $Union_{A1}$ → 20% $Union_B$ #14C. 80% $Union_{A2}$ → 20% $Union_B$	English	80% of *Union* dataset 80% of $Union_{A1}$ dataset 80% of $Union_{A2}$ dataset	27,833,899	20 % $Union_B$ dataset 20 % $Union_B$ dataset 20 % $Union_B$ dataset	10,785,542	*Basic* *Basic* *Basic*

- Evaluation



(a) #13: Tianya, 126 → Taobao

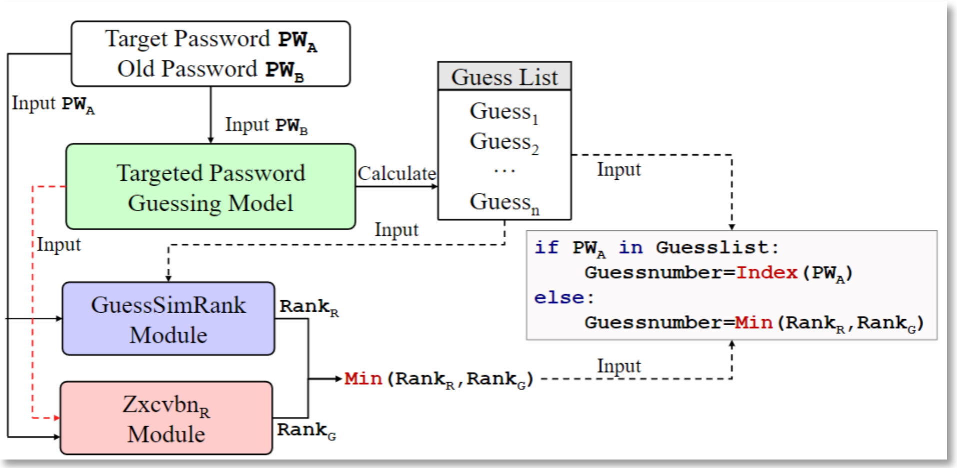


(b) #14: 80% *Union* → 20% *Union*

- Within 100 guesses, its average success rate in cracking is **17.20%** higher than PointerGuess in scenario #13 and **38.78%** higher in scenario #14.

Further exploration

- Employ POINTERGUESS to evaluate password strength



- Apply POINTERGUESS into C3 services

Generate a set of variants based on **IloveMacOP**

IloveMacOP	Iloveyou	MacOPIlove	IloveMacOP123	MacOP5789
IloveMacOP1	loveMacOP!	Ipassword1!	loveMac123	IloveOP5789

Thank you!

PointerGuess: Targeted Password Guessing Model using Pointer Mechanism

Kedong Xiu

Nankai University

kedongxiu@mail.nankai.edu.cn

On job market!

Ding Wang

Nankai University

wangding@nankai.edu.cn