Password Guessing Using Random Forest

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

Zijian Zhang
Peking University
zhangzj@pku.edu.cn

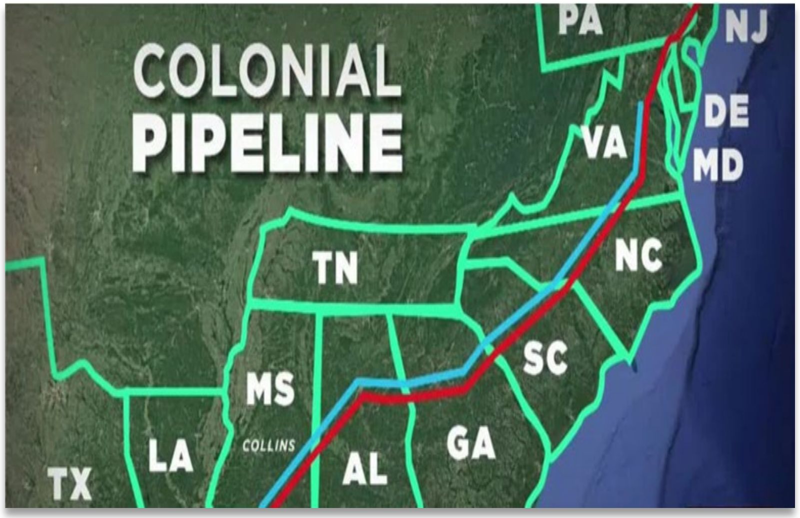
Kedong Xiu
Nankai University
kedongxiu@nankai.edu.cn

The 32nd USENIX Security Symposium

Passwords are ubiquitous

Billions of passwords leaked

- “Our dataset currently contains **953,894** incidents, of which **254,968** are confirmed breaches” [DBIR 2023].
- About **86%** of basic web application attacks were due to **stolen passwords**.
- **Poorly picked (weak) and protected** passwords continue to be one of the major sources of breaches.



The network of Colonial Pipeline breach

The celebrity photos leakage

5.6 million users’ fingerprint data breach

Password strength: resistance to guessing attempts

How much security strength can passwords actually provide?

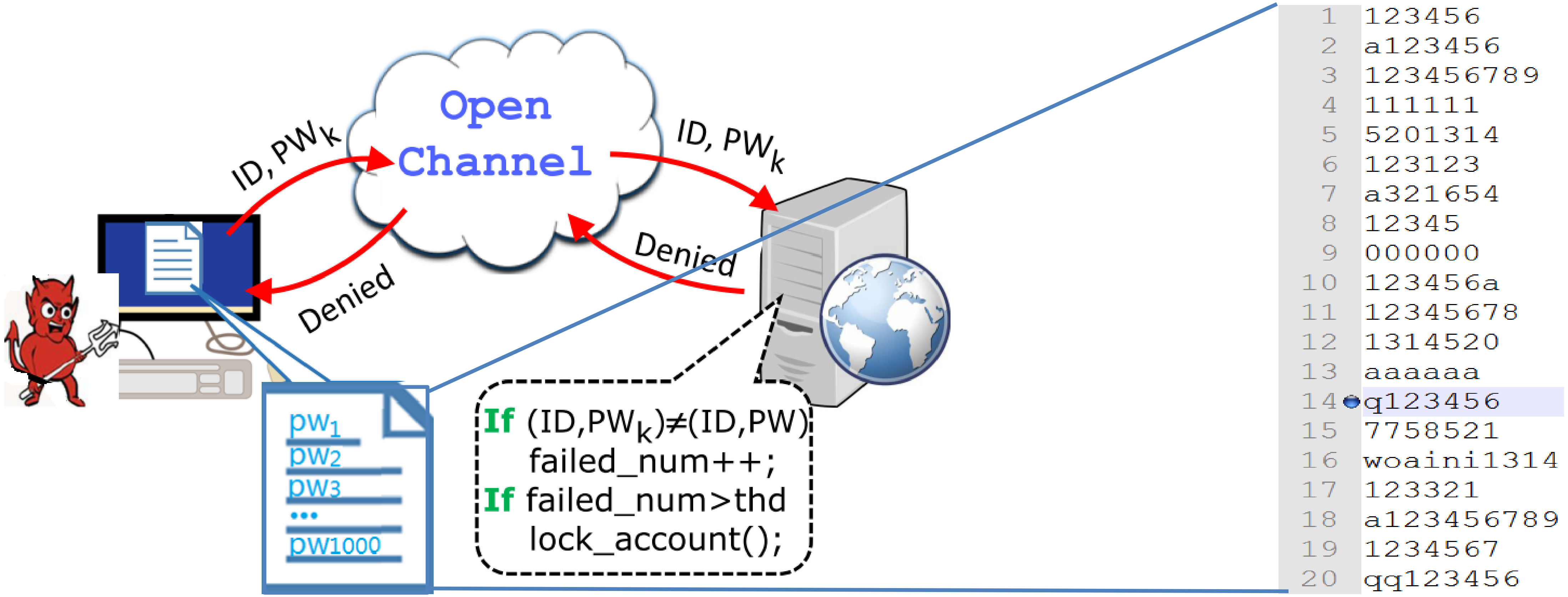
A more fundamental question

How to guess the user's password with the **least number of guesses?**

Password: the first line of defense against cyber attacks on a system.

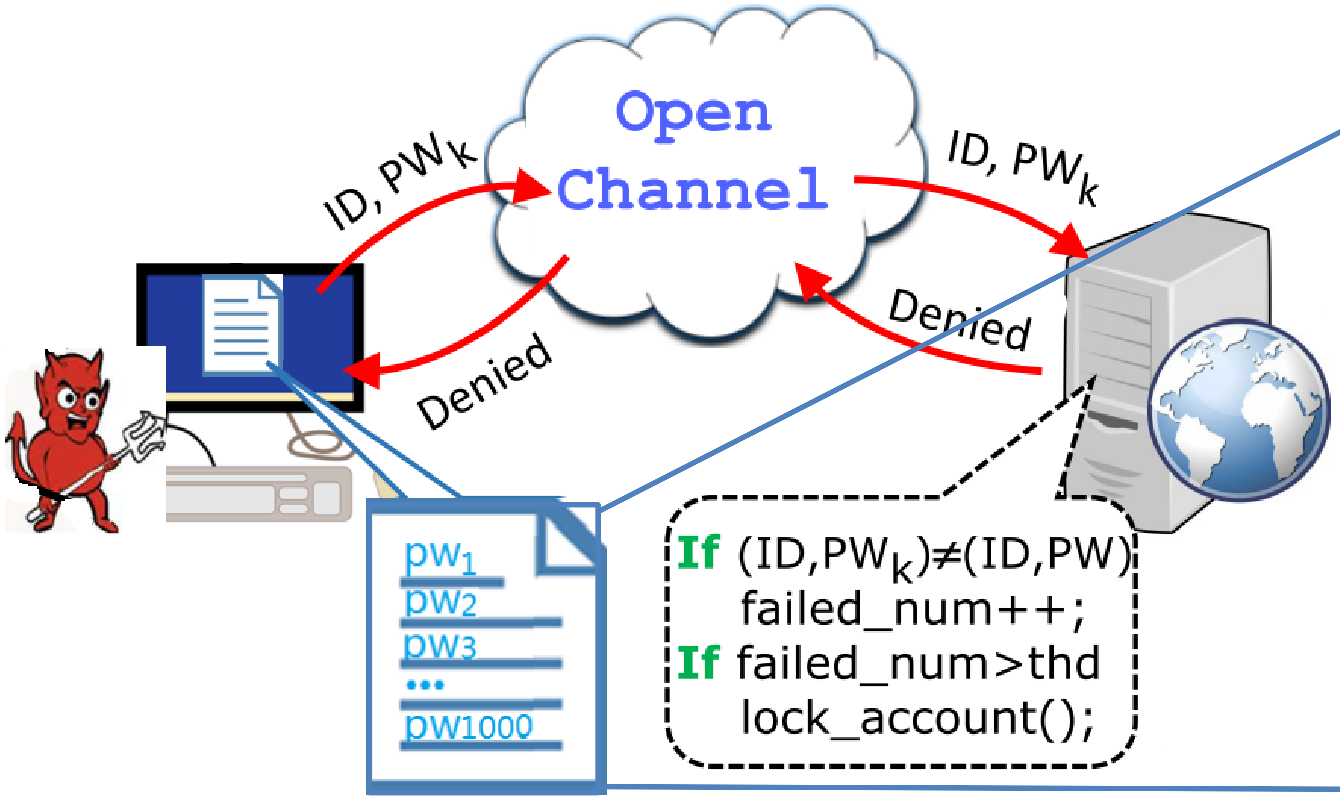
Password guessing scenarios

- **Trawling guessing**
  - The attacker generates **the same** password guessing dictionary <u>for all target users.</u>



Password guessing scenarios

- **Targeted guessing**
  - The attacker generates **a corresponding attack dictionary** for each target user.




```
B_0
E_0
A_2
N_1 B_0
N_0
A_0
B_6
N_1 B_6
123456
E_2
B_0 N_1
N_0 B_2
N_1 A_2
N_0 123
N_0 B_3
N_1 B_2
N_1 B_3
N_2 B_2
N_2 B_0
N_1 123456
```

Where is classical machine learning?

Probabilistic

- 2005 Markov [Narayanan-Shmatikov, ACM CCS 2005]
- 2009 PCFG [Weir et al., IEEE S&P 2009]
- 2014 Smoothing and regularization techniques [Ma et al., IEEE S&P 2014]

……

Deep-learning

- 2016 RNN [Melicher et al., USENIX Security 2016]
- 2019 PassGAN [Hitja et al., ACNS 2019]
- 2021 AdaMs [Pasquini et al., USENIX Security 2021]
- 2021 CPG/DPG [Pasquini et al., IEEE S&P 2021]
- 2021 Chunk-level [Xu et al. ACM CCS 2021]

……

Research on password guessing

Types of password models and typical representatives	Success rate	Efficiency	Interpre-tability	Proposed time
Statistical-based（PCFG，Markov）	Mid	High	High	2009-
Deep learning-based（RNN）	Mid	Low	Low	2016-
Classical machine learning (SVM)	Unknown	Mid?	Mid?	**Yet to be studied**

Research questions

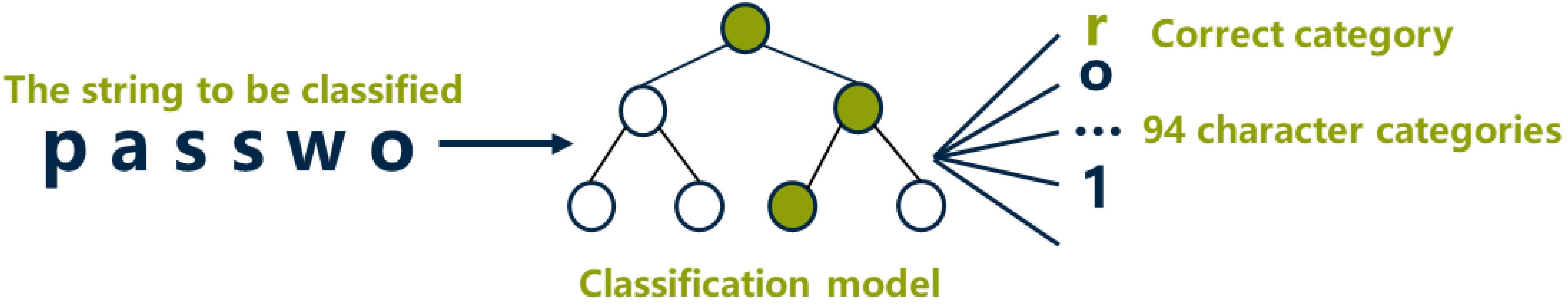
- Can **classical machine learning techniques** be used to design password models?
- If it is possible, **how can** these techniques be used for typical guessing scenarios?
- Whether password guessing models based on classical machine learning techniques can **improve the guessing success rate?**

Design challenges

- **Password guessing is different** from traditional NLP tasks. E.g., il0veu4ever (with the semantic love you forever);
- Cracking passwords requires **an exact match**: Any **vagueness** will not succeed. E.g., **P**@sswor123 and **p**@ssword123;
- How to **construct and select features** to ensure the effectiveness of machine learning algorithms?

Password guessing modeling

- Modeling password generation as a **Multi-Classification problem**
  - Our work makes **the same assumption** with the well-known Markov model: Each character in the password is only related to the previous characters.



Password feature construction

Feature construction method

- Each **character** is represented by **4-dimensional** features: (Character **type**, Character **serial number**, **Row number** of the keyboard, **Column number** of the keyboard)
- The entire n-order string uses **additional** 2 dimensions to represent the current length feature: (position of the character **in a password**, position of the character **in the current segment**)
- Each **6-order** string is represented as a **26** (=6×4+2) dimensional feature vector

q w e r 6 5 4 3 2 1

Length feature

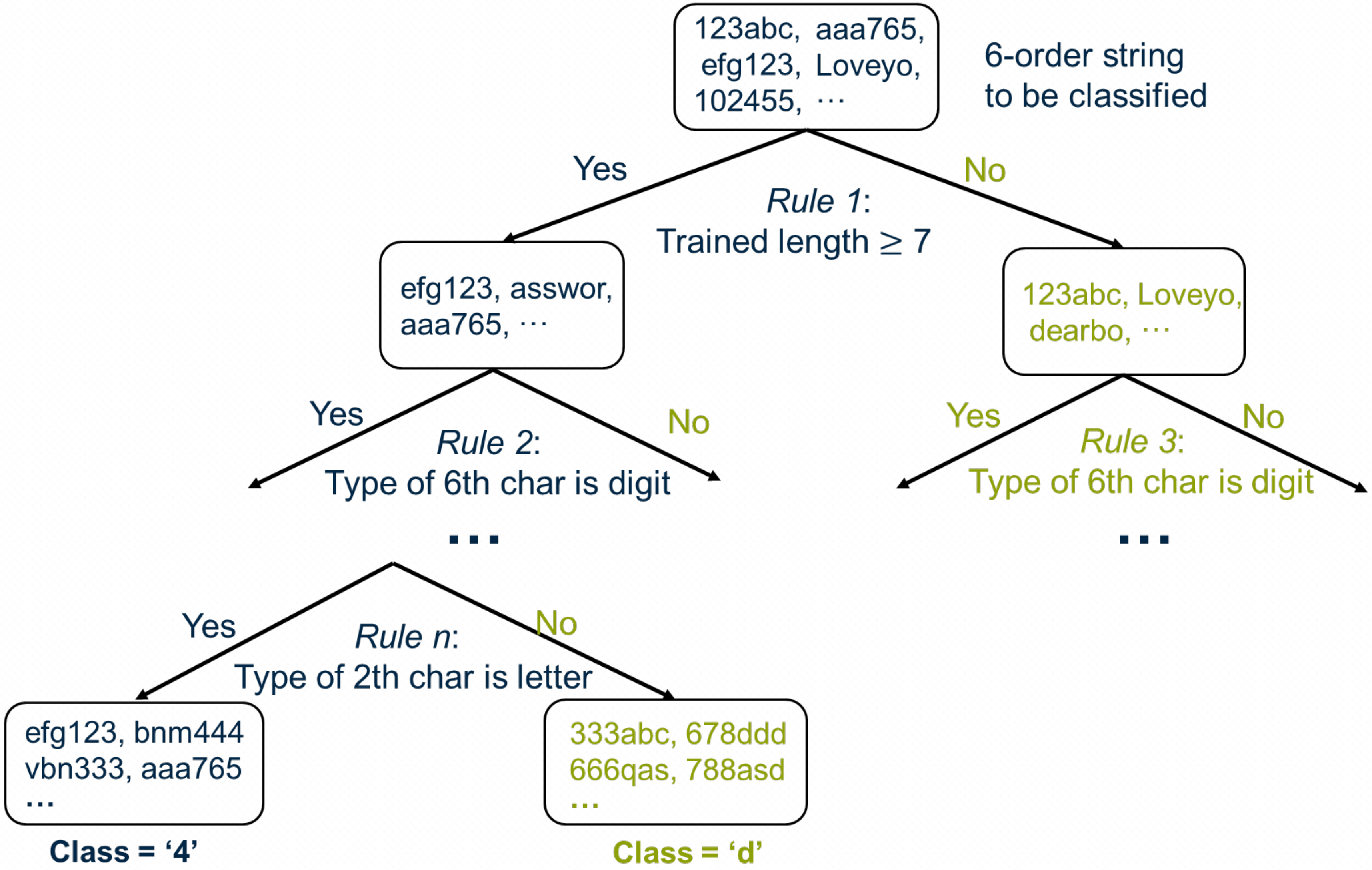
[(3, 23, 2, 2) (3, 5, 2, 3) (3, 18, 2, 4) (1, 6, 1, 6) (1, 5, 1, 5) (1, 4, 1, 4) (7, 3)]

(3, 23, 2, 2) = (Letter, **w** ranks 23rd among a~z, **w** is at **row 2** of the keyboard, **w** is at **column 2** of the keyboard)

(7, 3) = (length(**qwer654**), length(**654**))

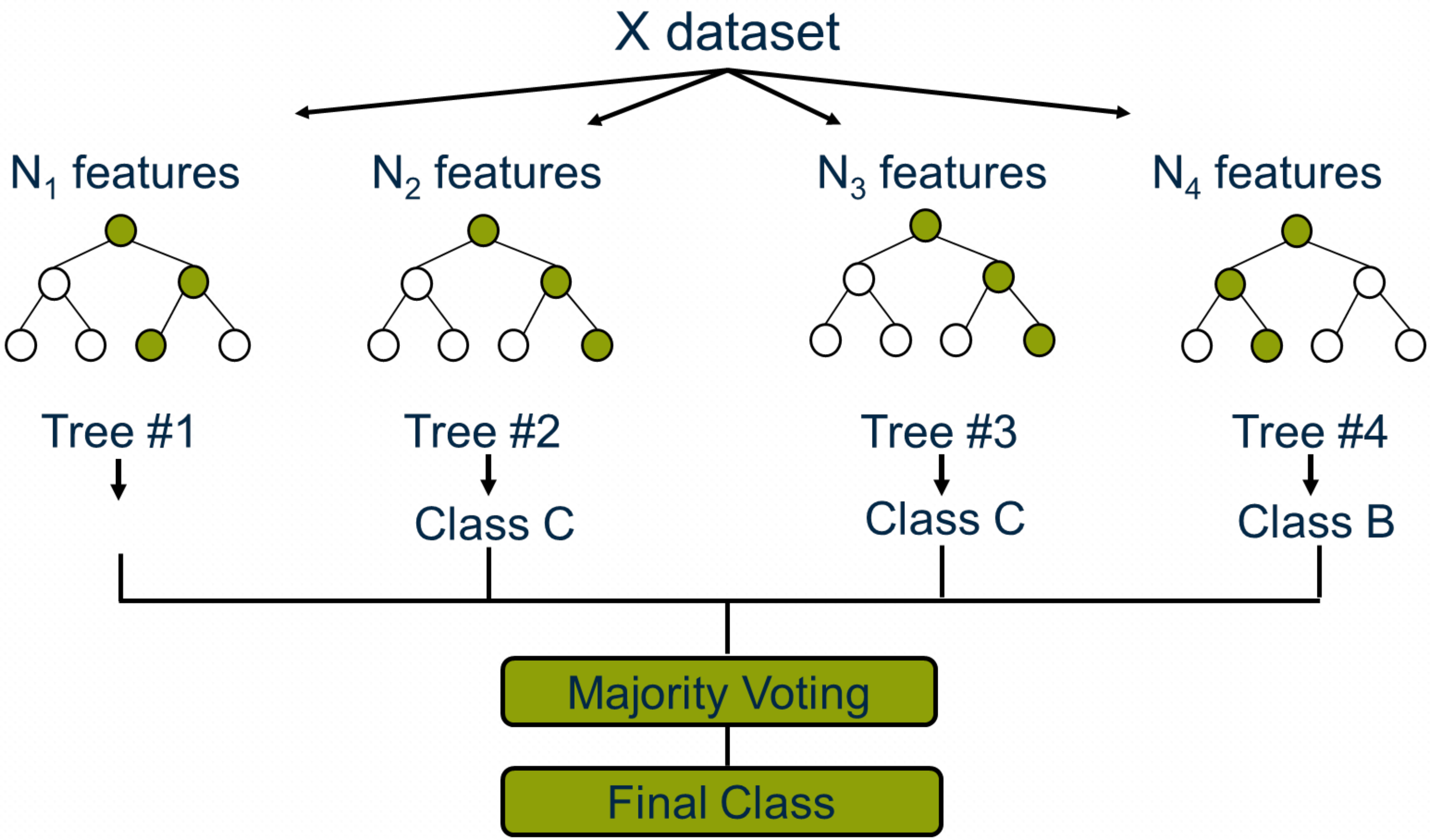
RFGuess: a trawling password model

- Use the **decision tree** for password prefix classification.



RFGuess: a trawling password model

- Vote on character classification results with **random forest**.
- The remaining password generation process **is the same as** the Markov model.



Experimental setup

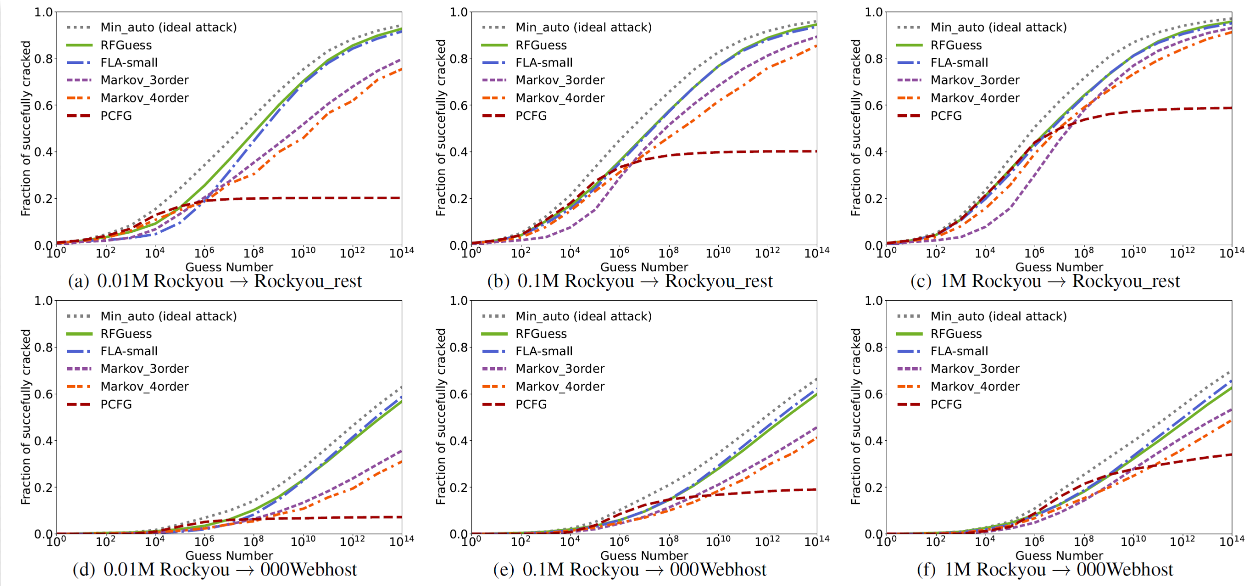
- 13 password datasets：5 Chinese datasets and 8 English datasets
  - Small-scale training set: 10,000, 100,000, and 1 million Rockyou
  - Large-scale training set: 75% of 000Webhost (~10 million)
- Two test scenarios: **intra-site** guessing and **cross-site** guessing scenarios

Table 1: Basic information about our 13 password datasets.†

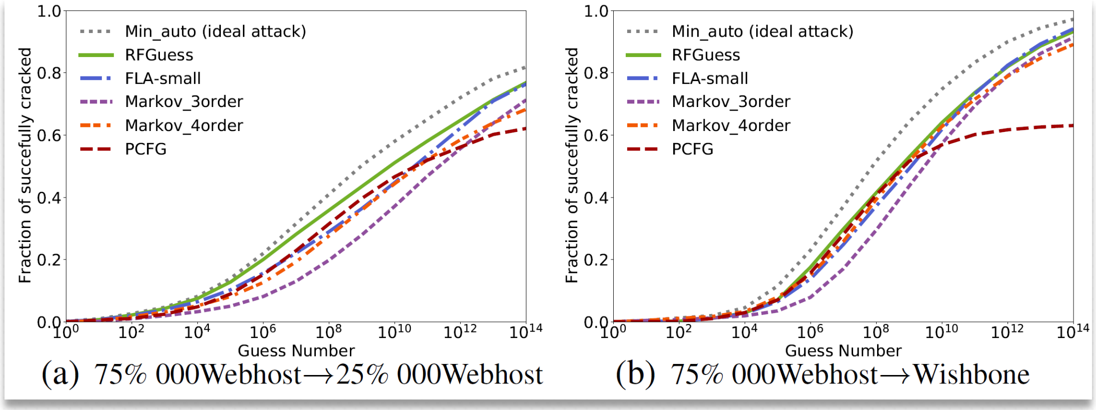
Dataset	Web service	Language	When leaked	Total PWs	Length>30	Removed %	Unique PWs	With PII
Taobao	E-commerce	Chinese	Feb., 2016	15,072,418	88	0.01%	11,633,759	
126	Email	Chinese	Oct., 2015	6,392,568	621	0.23%	3,764,740	
Dodonew	E-commerce	Chinese	Dec., 2011	16,283,140	13,4758	0.15%	10,135,260	
CSDN	Programmer	Chinese	Dec., 2011	6,428,632	0	0.01%	4,037,605	
Wishbone	Social	English	Jan., 2020	10,092,037	250	0.01%	5,933,902	
Mate1	Dating website	English	Mar., 2016	27,401,505	12,430	0.06%	11,916,080	
000Webhost	Web hosting	English	Oct., 2015	15,299,907	4,159	0.76%	10,526,769	
Yahoo	Web portal	English	July, 2012	453,491	0	2.35%	342,510	
LinkedIn	Job hunting	English	Jan., 2012	54,656,615	17,162	0.22%	34,282,741	
Rockyou	Social forum	English	Dec., 2009	32,603,387	3,140	0.07%	14,326,970	
12306	Train ticketing	Chinese	Dec., 2014	129,303	129,303	0	117,808	✓
ClixSense	Paid task platform	English	Sep., 2016	2,222,045	0	0	1,628,018	✓
Rootkit	Hacker forum	English	Feb., 2011	69,330	5	0.01%	56,835	✓

†PW stands for password, and PII for personally identifiable information. We clean up passwords longer than 30 and containing non-ASCII codes.

Experimental results



(a) 0.01M Rockyou → Rockyou_rest (b) 0.1M Rockyou → Rockyou_rest (c) 1M Rockyou → Rockyou_rest
(d) 0.01M Rockyou → 000Webhost (e) 0.1M Rockyou → 000Webhost (f) 1M Rockyou → 000Webhost



(a) 75% 000Webhost→25% 000Webhost (b) 75% 000Webhost→Wishbone

Table 7: Performance of different models.†

Model	RFGuess	PCFG [69]	3-order Markov [42]	FLA [43]
Training time	0.3h	24s	102s	16h
Model size	4.5G	93.2M	1.4G	5.8M
Generated PW/s	130	82,372	13,303	2,500

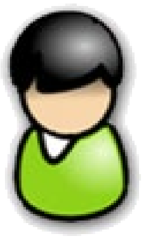
† CPU: Xeon silver 4210R 2.4GHz; GPU: GeForce RTX 3080 (5M dataset).

- RFGuess achieves a guessing success rate **comparable** to deep learning-based methods (FLA) and **outperforms** other statistical-based guessing methods.
- RFGuess suffers from the drawbacks of slow password generation speed and high memory consumption.

More suitable for online password guessing

RFGuess-PII: a targeted password model

- PII matching disambiguation



ID: wang123@foo.com ;
name: Wang Lei; birthday: 1980.01.23

wang1231980 → $\mathbf{N_1 123 B_2}$ or $\mathbf{U_1 B_2}$ or $\mathbf{N_1 B_7}$ **Which one to choose?**

- Optimized PII matching algorithm
  - We propose a PII matching algorithm based on the principle of minimum information entropy

PW1: R1 R2 R3

PW2: R1 R2 R4

PW3: R1 R5

PW4: R2 R3

PW5: R1 R8 R9

1. Exhaustively enumerate **all possible representations** for all passwords;
2. Count all representations, **sort globally by frequency**, and take out the representation with the most frequency as the priority representation (**such as R1**);
3. Update the frequency, and then **take out the representation with the most frequency** among the remaining representations, as the second priority representation (**such as R2**), and **iterate until the frequency of all representations is 1.**

Password feature construction (PII)

- The feature construction method **is similar to** RFGuess
- The **differences** lies:
  - A string containing personal information is regarded as a **PII segment.**
  - E.g., Wang.1980: **Wang** and **1980** are each regarded as a complete **segment**, represented by four-dimensional features: **(personal information type, personal information serial number, 0, 0).**
  - Here the last two **0**s are to **align with** the feature of ordinary characters.

Bs Bs wang 1 2 3 1980

Length feature

[(0, 0, 0, 0) (0, 0, 0, 0) (10, 1001, 0, 0) (1, 1, 1, 1) (1, 2, 1, 2) (1, 3, 1, 3) (4, 3)]

Bs = Beginning symbol **A PII segment** **An ordinary character**

Datasets and experimental setup (PII)

- Dataset: 6 password datasets, including **4~6 kinds of PII**

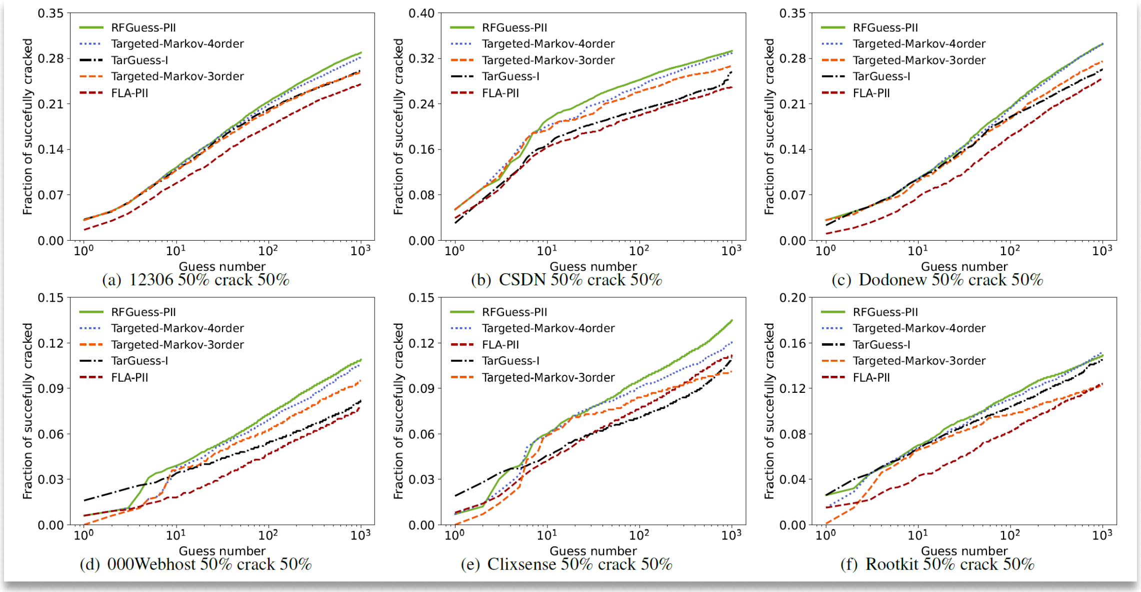
Table 2: Basic information about our PII datasets.

Dataset	Language	Items num	Types of PII useful for this work
12306	Chinese	129,303	Email, User name, Name, Birthday, Phone
CSDN	Chinese	77,216	Email, User name, Name, Birthday, Phone
Dodonew	Chinese	161,517	Email, User name, Name, Birthday, Phone
ClixSense	English	2,222,045	Email, User name, Name, Birthday
000Webhost	English	79,580	Email, User name, Name, Birthday
Rootkit	English	69,418	Email, User name, Name, Birthday

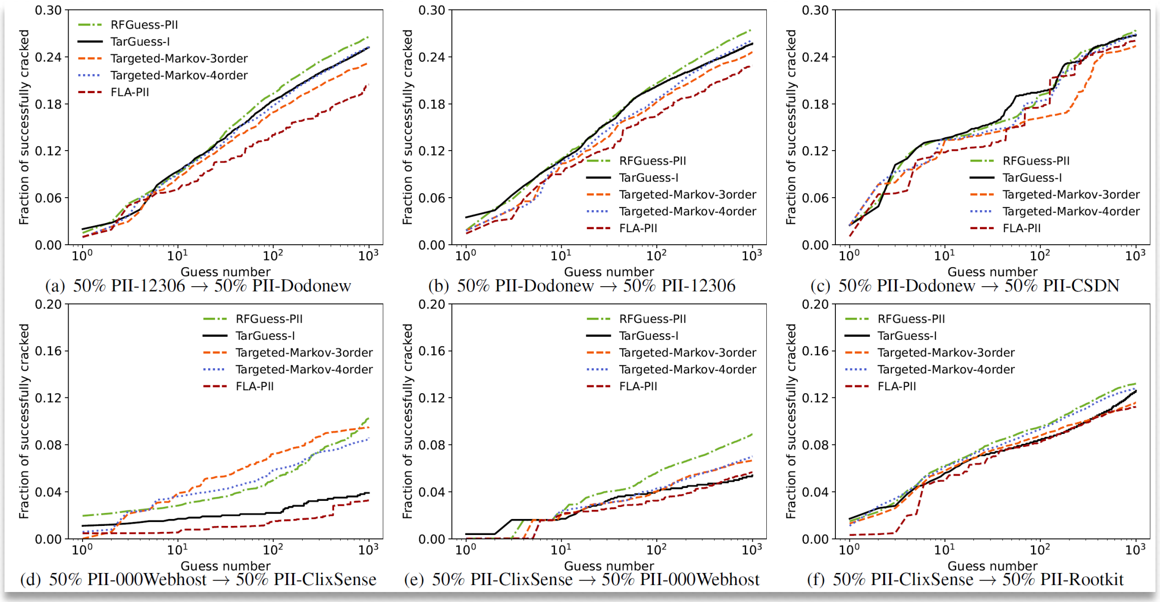
- Experimental setup
  - **Intra-site** guessing scenarios: e.g., 50% PII-12306→50% PII-12306
  - **Cross-site** guessing scenarios: e.g., 50% PII-12306→50% PII-Dodonew

Experimental results (PII)

- Within 100 guesses, the guessing success rate of RFGuess-PII is **20%~28%**;
- RFGuess-PII outperforms existing models by **7%~13%** within **1,000 guesses**.



Intra-site guessing scenarios

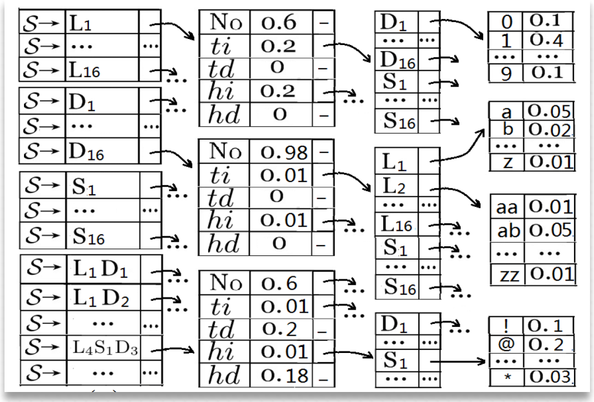


Cross-site guessing scenarios

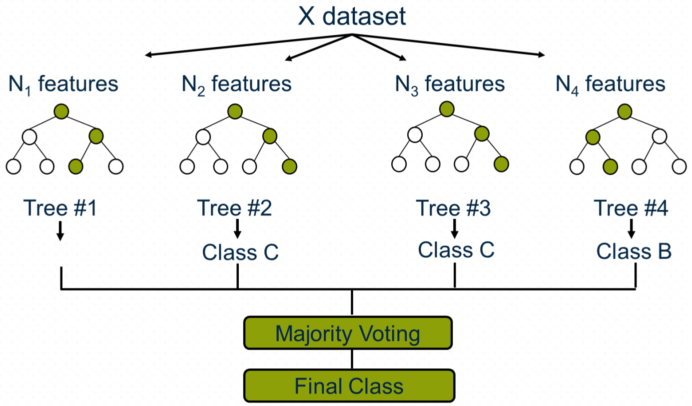
RFGuess-Reuse: a reuse model

Username	Password
zhangsan	abc334bca Abc334bca123
lisi001	Qwerdf 123456qwerdf
…	…

Users' password pairs



Count the **structure-level** operations of password pairs in the train set (e.g., L8D5→L7S2)



Predicting the **segment-level** operations using the **random forest model** (e.g., passwor→password)

PW1 = abc334bca

$$\Pr(pw_1 \rightarrow pw_2) = \left(\prod_{i=1}^{n} \Pr(Pt^{i}_{pw_1 \rightarrow pw_2})\right) * p_n$$

Guesses	Prob.
abc334bca1	0.6
abc334bca123	0.2
abc34	0.1
…	…

Datasets and experimental setup (Reuse)

- Dataset: 8 datasets containing **password pairs** (obtained through **email** match)

Table 4: Basic information about password reuse datasets.

Dataset	Language	Items	# Same password pair	# Similar password pair†
CSDN→126	Chinese	195,832	62,686	47,690
CSDN→12306	Chinese	12,635	7,079	2,815
12306→Dodonew	Chinese	49,775	35,395	9,386
CSDN→Dodonew	Chinese	5,997	2,040	1,597
000Webhost→Clixsense	English	150,273	35,470	41,731
000Webhost→LinkedIn	English	231,452	50,875	52,731
000Webhost→Yahoo	English	36,936	5,960	6,303
000Webhost→Mate1	English	51,942	7,613	25,504

† Similar means the similarity score is within [0.5, 1.0], and it is calculated as $s = 1 - \text{EditDistance}(pw1, pw2)/\max(|pw1|, |pw2|)$.

- Experimental setup
  - ***A*** → ***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***.
  - CSDN → 126 is the training set for Chinese attack scenarios.
  - 000Webhost → ClixSense is the training set for English attack scenarios.

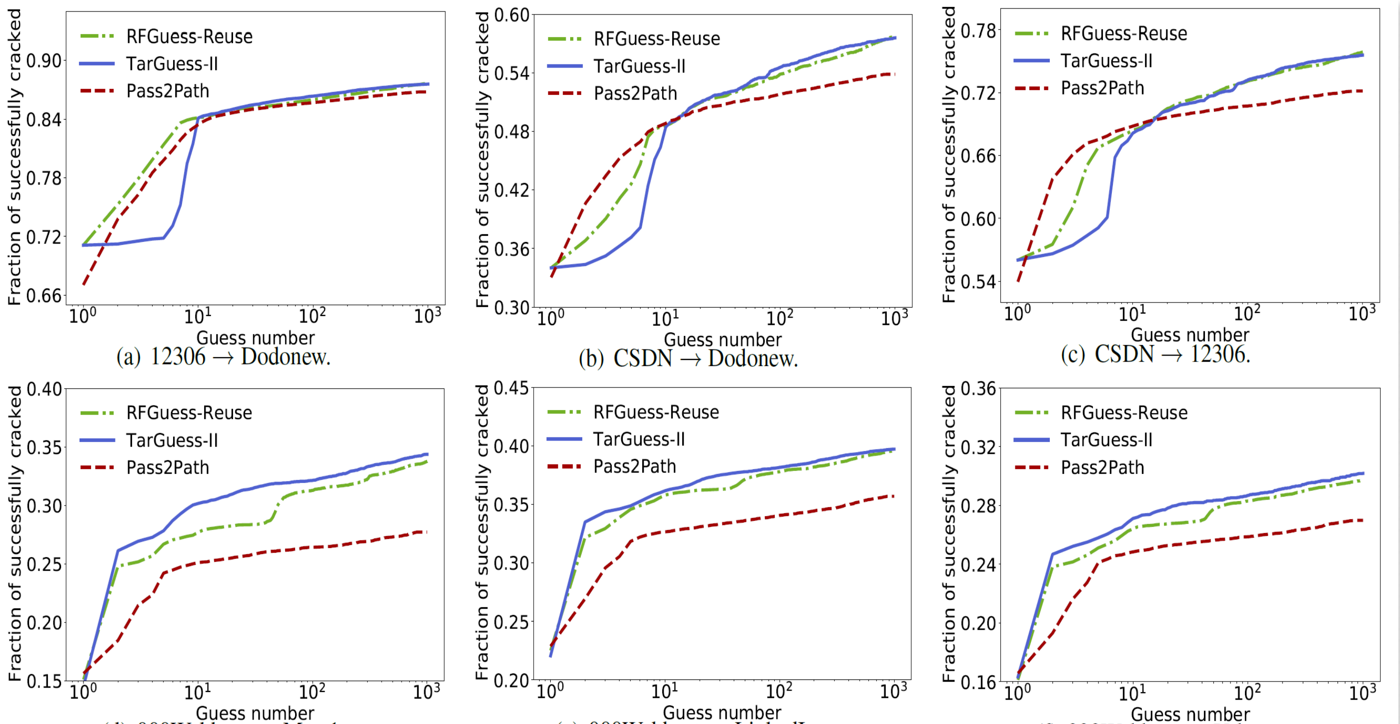
Experimental results (Reuse)

- RFGuess-Reuse is **comparable** to existing leading models within 1,000 guesses

Table 5: Comparison of three password reuse models.†

Experimental setup		RFGuess-Reuse	Pass2-path [45]	TarGuess-II [64]
Guessing scenario	Guess number			
CSDN → 12306	10	68.41%	68.80%	68.13%
	100	73.09%	70.72%	73.19%
	1,000	75.86%	72.16%	75.57%
CSDN → Dodonew	10	48.59%	48.82%	48.44%
	100	53.86%	51.79%	54.56%
	1,000	57.71%	53.84%	57.58%
12306 → Dodonew	10	84.14%	83.44%	84.11%
	100	86.00%	85.69%	86.34%
	1,000	87.65%	86.78%	87.58%
000webhost → Mate1	10	27.70%	25.11%	30.17%
	100	31.29%	26.42%	32.14%
	1,000	33.77%	27.73%	34.37%
000webhost → LinkedIn	10	35.67%	32.65%	36.17%
	100	37.77%	34.06%	38.16%
	1,000	39.52%	35.69%	39.72%
000webhost → Yahoo	10	26.53%	24.84%	27.12%
	100	28.59%	25.87%	28.69%
	1,000	30.13%	26.99%	30.19%

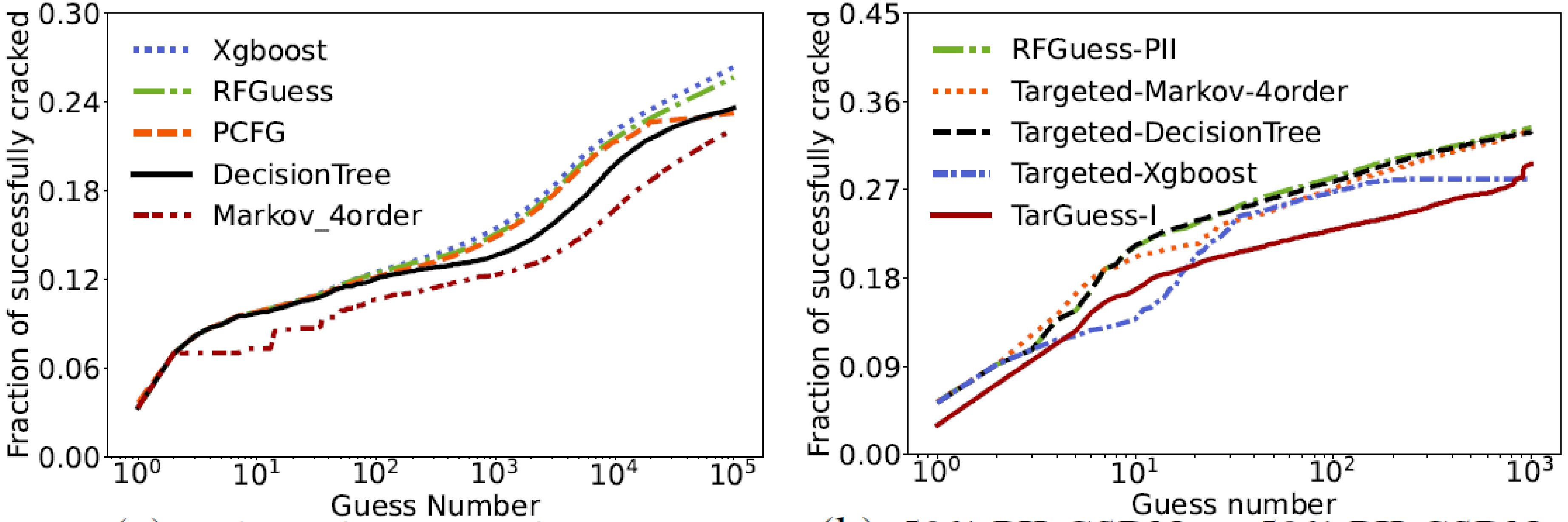
†A value with dark gray (resp. light gray) represents the highest one (resp. 2nd one).



(a) 12306 → Dodonew. (b) CSDN → Dodonew. (c) CSDN → 12306.
(d) 000Webhost → Mate1. (e) 000Webhost → LinkedIn. (f) 000Webhost → Yahoo.

General applicability

- Our password character encoding method is **applicable to a series of supervised algorithms** that can tackle multi-classification problems.
- Among these supervised algorithms, **boosting method** performs well.



(a) 0.5M CSDN $\rightarrow$ CSDN_rest (b) 50% PII-CSDN $\rightarrow$ 50% PII-CSDN

Thank you!

Password Guessing Using Random Forest

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

Zijian Zhang
Peking University
zhangzj@pku.edu.cn

Kedong Xiu
Nankai University
kedongxiu@nankai.edu.cn

The 32nd USENIX Security Symposium