Incrementally Updateable Honey Password Vaults

Haibo Cheng[1], Wenting Li[1],
Ping Wang[1], Chao-Hsien Chu[2], Kaitai Liang[3]

[1] *Peking University,* [2] *Pennsylvania State University,* [3] *Delft University of Technology*

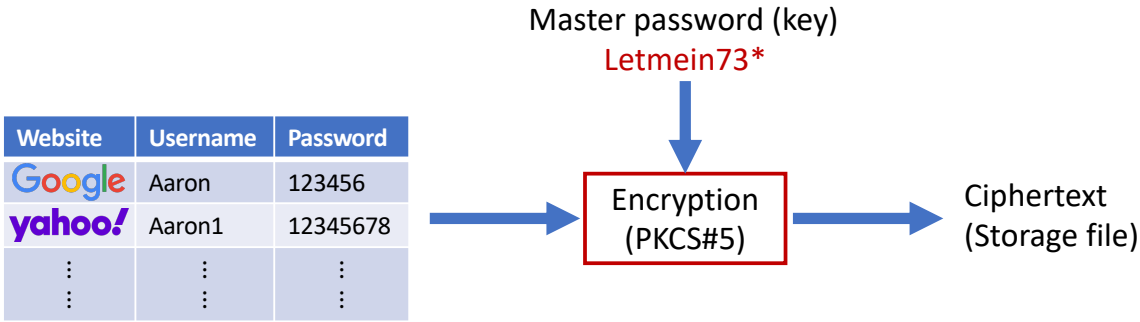
August 11, 2021 @ USENIX Security

PEKING
UNIVERSITY

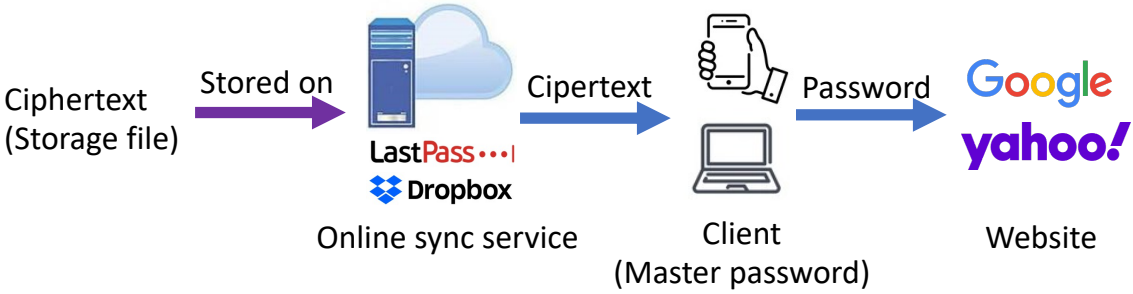
PennState

TU Delft Delft University of Technology

Password vaults



Password vaults



Traditional password vaults suffer from offline guessing

Online sync service

Steal

Ciphertext

Decrypt with

Correct guess

Password
123456
12345678
⋮

✓

Incorrect guesses

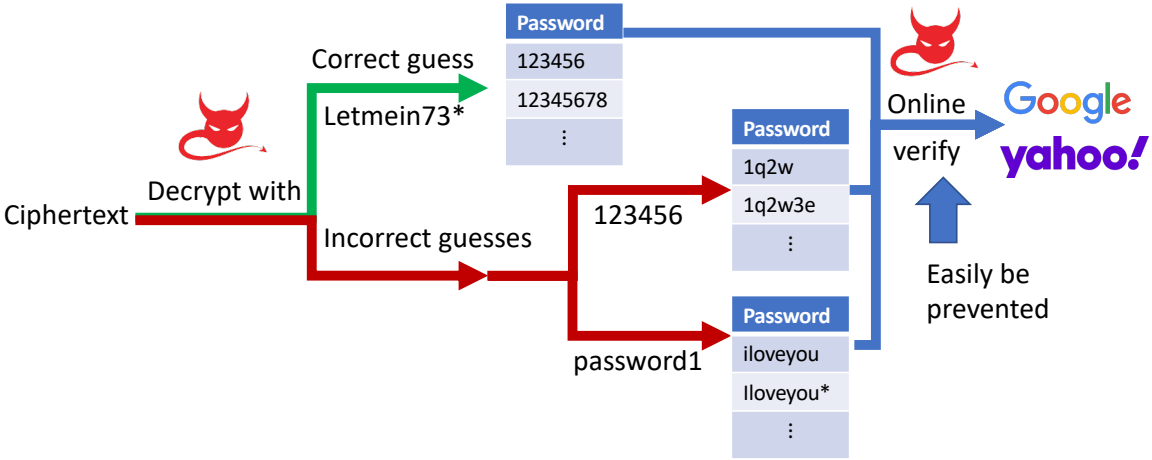
Failures or junk

Φб ЈӇ•˚₦ЌѰ

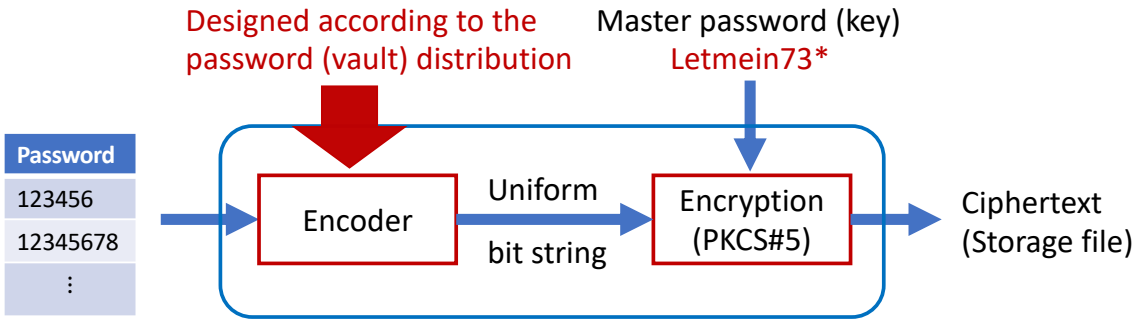
✗

Master passwords are human-memorable and may be easily guessed

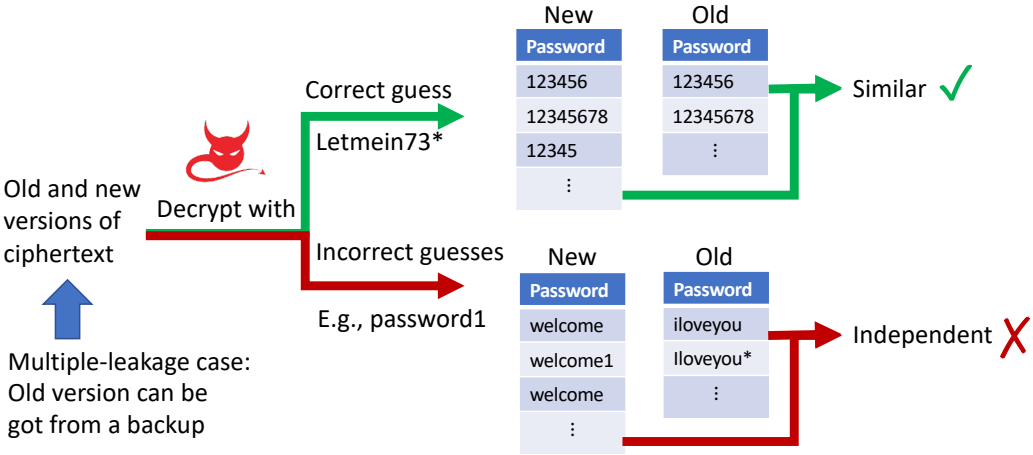
Honey password vaults



Honey password vaults: Design



An open problem for honey password vaults: How to achieve update security if the user adds or changes a website password



Our contribution

1. New designs:
   a. A generic construction and an incremental update mechanism, achieving update security.
   b. An instantiation of the construction, generating more plausible-looking decoys.
2. Security evaluation:
   a. Formally investigate the optimal strategy for distinguishing decoys and further propose practical attacks.
   b. Evaluate the current and our designs with the attacks.

A generic construction for password vaults

Probability model

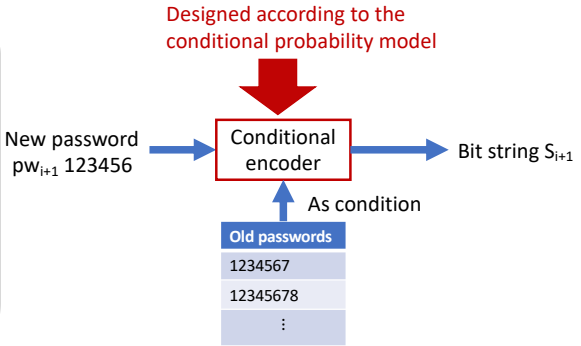
$$\Pr_{\text{real}}(V) = \prod_{i=0}^{n-1} \Pr_{\text{real}}(pw_{i+1} \mid pw_1, pw_2, \ldots, pw_i). \tag{1}$$

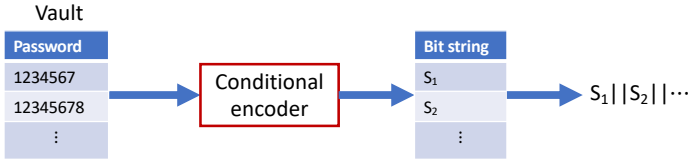
1. Basic idea: A user generates the passwords one by one. $\Pr_{\text{real}}(pw_{i+1} \mid pw_1, pw_2, \ldots, pw_i)$ is the conditional probability that the user generates pw_{i+1} under given old passwords $(pw_1, pw_2, \ldots, pw_i)$.
2. Our design: We use a conditional probability model $\Pr_{\text{MSPM}}(\cdot|\cdot)$ (multi-similar-password model) to estimate $\Pr_{\text{real}}(\cdot|\cdot)$.

A generic construction for password vaults

Conditional encoder

1. Encode a new password pw_{i+1} given old passwords $(pw_1, pw_2, \ldots, pw_i)$.
2. Designed according to the conditional probability model $\Pr_{\mathrm{MSPM}}(\cdot|\cdot)$ by Cheng et al.'s transformation [1].
3. Encode a vault password by password.





An incremental update mechanism for password vaults

Encoder+Encryption

1. Encoder: The conditional encoder.
2. Encryption: A prefix-keeping scheme, e.g., CTR-mode AES with PBKDF.

Old vault

Website	Username	Password index
Google	Aaron	1
yahoo!	Aaron1	2
⋮	⋮	⋮

Password
1234567
12345678
⋮

Bit string
S_1
S_2
⋮

Add a password

Website	Username	Password index
Google	Aaron	1
yahoo!	Aaron1	2
⋮	⋮	⋮
facebook	Aaron	i+1

Password
1234567
12345678
⋮
12345

Bit string
S_1
S_2
⋮
S_{i+1}

An incremental update mechanism for password vaults

Old vault

Website	Username	Password index
Google	Aaron	1
yahoo!	Aaron1	2
⋮	⋮	⋮

Password
1234567
12345678
⋮

Bit string
S_1
S_2
⋮

Delete a password

Website	Username	Password index
Google	Aaron	1
yahoo!	Aaron1	Deleted
⋮	⋮	⋮

Password
1234567
12345678
⋮

Bit string
S_1
S_2
⋮

Change a password

Website	Username	Password index
Google	Aaron	1
yahoo!	Aaron1	i+1
⋮	⋮	⋮

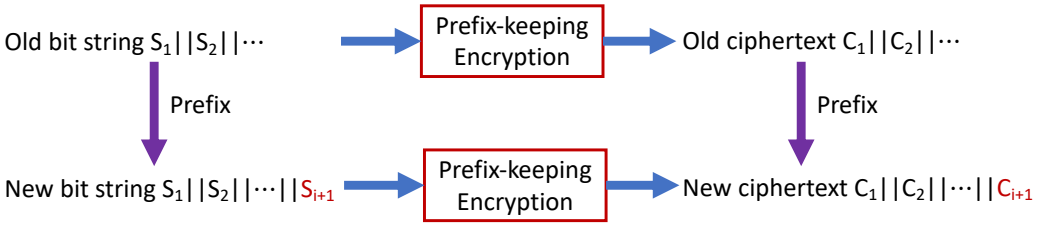
Password
1234567
12345678
⋮
12345

Bit string
S_1
S_2
⋮
S_{i+1}

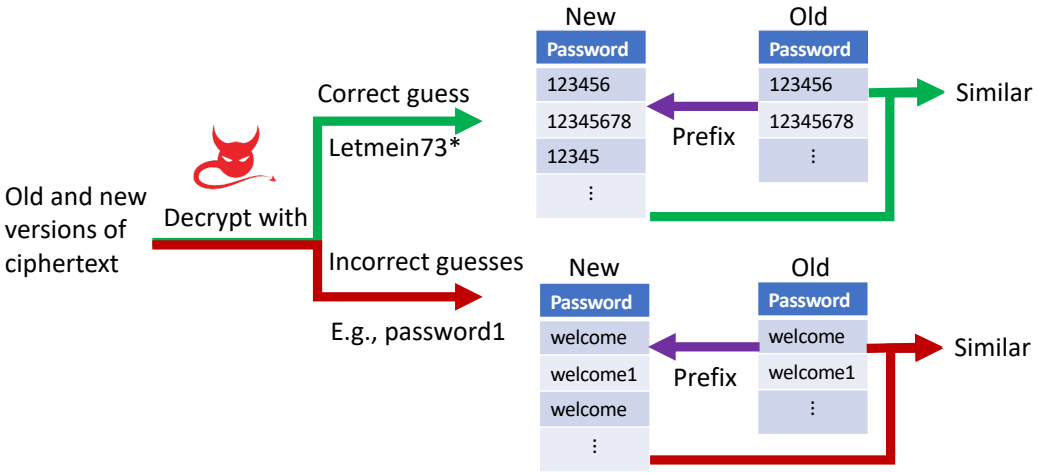
An incremental update mechanism for password vaults

Security

1. Old ciphertext is a prefix of the new one.
2. The attacker with the two versions of the ciphertext (multi-leakage case) degenerates to an attacker only with the current version (single-leakage case).



Our incremental update mechanism achieves update security



A new conditional probability model for password vaults

1. Basic idea: A user generates a new password by 1) reusing an old one or 2) not (i.e., creating a brand new one).
2. Construction:

$$\begin{aligned} &\Pr_{\mathrm{MSPM}}(pw_{i+1} \mid pw_1, pw_2, \ldots, pw_i) \\ =&\frac{1-f(i)}{i} \sum_{i'=1}^{i} \Pr_{\mathrm{SSPM}}(pw_{i+1} \mid pw_{i'}) + f(i) \Pr_{\mathrm{SPM}}(pw_{i+1}). \end{aligned} \tag{2}$$

3. Instantiation:
   a. Single-similar-password model $\Pr_{\mathrm{SSPM}}$: our simple design
   b. Single-password model $\Pr_{\mathrm{SPM}}$: Markov [2]
   c. Unreused probability f: Nonlinear regression $f(i) = 1/(\sum_{k=0}^{3} a_k i^k)$

How to evaluate the indistinguishability of decoys

The optimal strategy of distinguishing real and decoy vaults

1. Ranking by the conditional probability of being real under the given ciphertext.
   The conditional probability for a vault V_i is proportional to the real-to-decoy probability ratio
   $$\frac{\Pr_{\text{real}}(V_i)}{\Pr_{\text{decoy}}(V_i)}.$$

Practical attacks

1. Cannot precisely calculate $\Pr_{\text{real}}$.
2. Estimate the ratio on the single password distribution and password-reuse features.

Evaluating the exiting and our honey vault schemes

Experimental results

1. Our attack is more effective than the state-of-the-art attack (KL divergence attack [3]) against the existing schemes.
2. Our design is brings 2.8x-7.5x online cost to attackers.

Table 1: The distinguishing accuracy of attacks against the honey vault schemes

Scheme	KL divergence attack [3]	Our attack
Chatterjee et al.'s [4]	86%	94%
Golla et al.'s [3] (static, 10^0)	52%	86%
Our design	58%	58%
Perfect design	50%	50%

Future work

1. The incremental update mechanism may apply to other applications using honey encryption.
2. The probability model design for passwords and password vaults may be used for password guessing.

Q&A

Thank you

References I

[1] Haibo Cheng et al. "Probability Model Transforming Encoders Against Encoding Attacks". In: *USENIX Security 2019*, pp. 1573–1590.

[2] Jerry Ma et al. "A Study of Probabilistic Password Models". In: *IEEE S&P 2014*, pp. 538–552.

[3] Maximilian Golla, Benedict Beuscher, and Markus Dürmuth. "On the security of cracking-resistant password vaults". In: *ACM CCS 2016*, pp. 1230–1241.

[4] Rahul Chatterjee et al. "Cracking-resistant password vaults using natural language encoders". In: *IEEE S&P 2015*, pp. 481–498.