Conditional Encryption with Applications to Secure Personalized Password Typo Correction

Mohammad Hassan Ameri
mameriek@purdue.edu

Jeremiah Blocki
jblocki@purdue.edu

PURDUE UNIVERSITY
Department of Computer Science

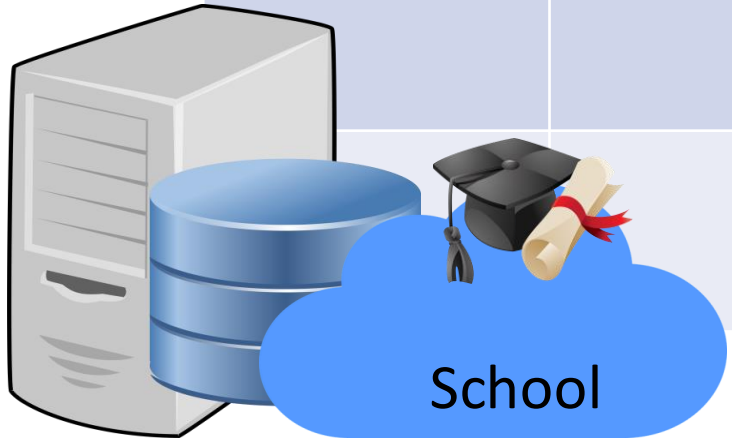
ACM SIGSAC Conference on Computer and Communications Security, 2024

Motivating Application: Password Typos

Resister: SpogeBob, password = Patrick123school

Registration

User ID	Salt Value	Hash (.)
SpongeBob	1010100000	4e0986f0bdeadc1ba347cb469b1ad55e68f36efc1efbd17319e43065d15160ce



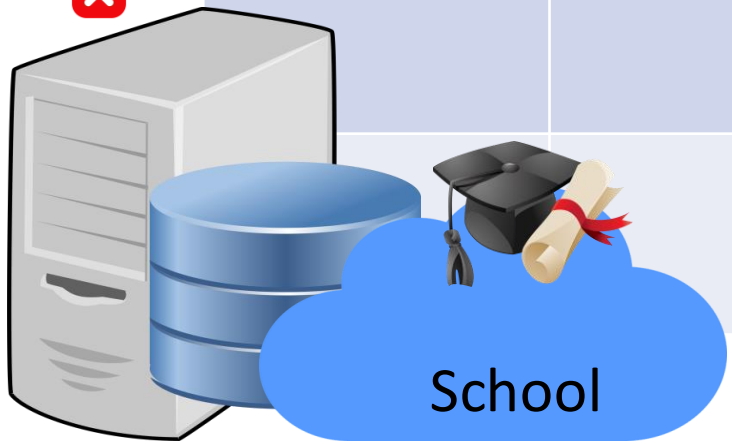
Motivating Application: Password Typos

Login: SpongeBob, password = pATRICK123SCHOOL ❌

Login: SpongeBob, password = Patrick1234school ❌

Login: SpongeBob, password = Patrick123School ❌

User ID	Salt Value	Hash (.)
SpongeBob	1010100000	4e0986f0bdeadc1ba347cb469b1ad55e68f36efc1efbd17319e43065d15160ce



Motivating Application: Password Typos

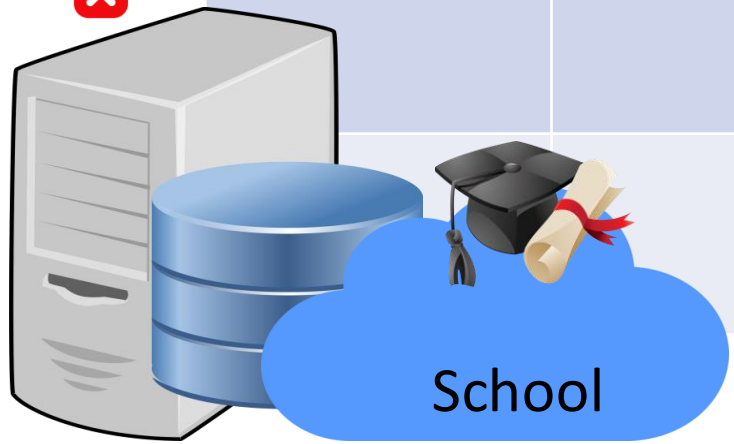
Login: SpongeBob, password = pATRICK123SCHOOL ❌

Login: SpongeBob, password = Patrick1234school ❌

Login: SpongeBob, password = Patrick123School ❌

Incorrect Login Attempts

User ID	Salt Value	Hash (.)
SpongeBob	1010100000	4e0986f0bdeadc1ba347cb469b1ad55e68f36efc1efbd17319e43065d15160ce



Motivating Application: Password Typo Correction

Prior Work:

Improving usability with *Relaxed Checking* mechanism:

- [CAAJT:S&P16]: **20%** of typos can be corrected by a small set of corrector functions. ✅

Further usability improvement**:** *Personalized password* typo correction

- [CWPCR:CCS17]: **45%** of users would benefit from password-typo personalization ✅

Question 🤔: Does typo tolerance make it easier for attacker to crack password?

- Answer [CWPCR:CCS17]: **No**, as long as we are careful about what typos are allowed ✅

Question 🤔: How do we learn what typos are common for each user?

[CAAJT:S&P16] Chatterjee, et al. "pASSWORD tYPOS and how to correct them securely." IEEE-S&P, 2016.
[CWPCR:CCS17] Chatterjee, et al. "The typtop system: Personalized typo-tolerant password checking." *ACM-CCS,* 2017**.**

Naïve Solution: Plaintext Typo Storage

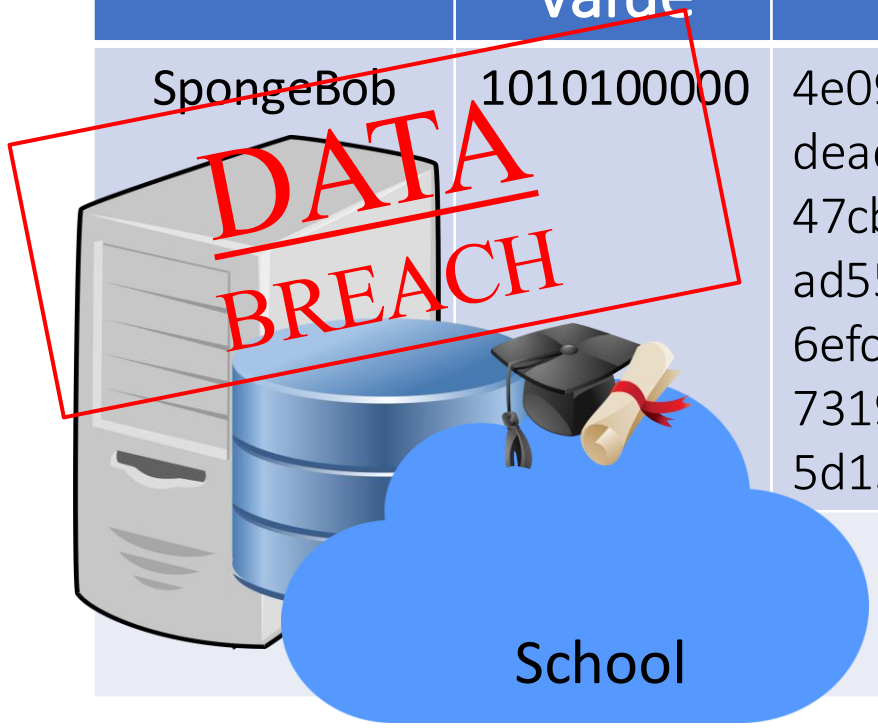
- **Problem:** Incorrect logins might help attack guess real password.

pATRICK123SCHOOL

Probably CAPSLOCK was ON!

Orig PWD: Patrick123school

User ID	Salt Value	Hash (.)	Incorrect Login Attempts (Plaintext)
SpongeBob	1010100000	4e0986f0b deadc1ba3 47cb469b1 ad55e68f3 6efc1efbd1 7319e4306 5d15160ce	pATRICK123SCHOOL Patrick1234school Patrick123School



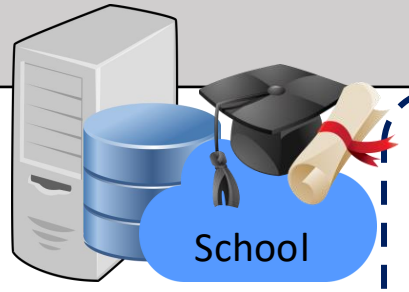
Improved Solution: TypTop [CWPCR:CCS17]

Resister: SpongeBob,
PWD = Patrick123school

Registration

Cache	Incorrect Login Attempts (Encrypted)
$C_{sk,PWD}$	---------

	---------
pk SpongeBob	

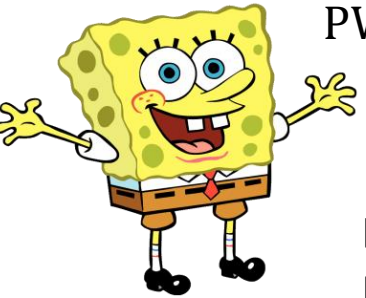


$(sk, pk) \leftarrow \mathsf{KG}(1^{\lambda})$

$K = \mathsf{PBKDF}(PWD)$

$C_{sk,PWD} = \mathsf{Auth}.\mathsf{Enc}_K(sk)$

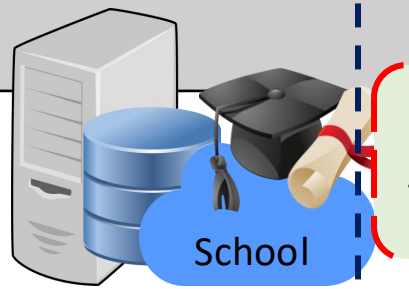
Improved Solution: TypTop [CWPCR:CCS17]



Login: SpongeBob
PWD_1 = pATRICK123SCHOOL

Login: SpongeBob
PWD = Patrick123school

Cache	Incorrect Login Attempts (Encrypted)
$C_{sk,PWD}$ C_{sk,PWD_1} pk SpongeBob	



$c_{PWD_1} = \mathrm{Enc}_{pk}(PWD_1)$

$K = \mathrm{PBKDF}(PWD)$

$sk = \mathrm{Auth.Dec}_K(c_{sk,PWD})$

Authentication is Approved and the server knows orig. PWD and sk

$PWD_1 = \mathrm{Dec}_{sk}(c_{PWD_1})$

$K_1 = \mathrm{PBKDF}(PWD_1)$

$C_{sk,PWD_1} = \mathrm{Auth.Enc}_{K_1}(sk)$

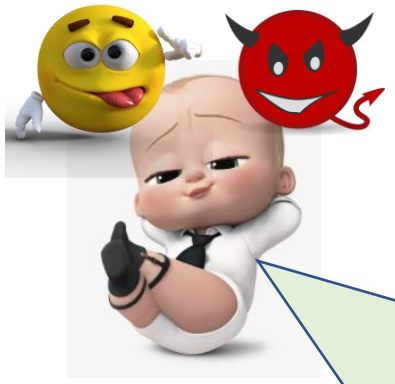
`` $PWD_1$ = **pATRICK123SCHOOL''** and ``**Patrick123school''** are close (CPASLOCK error), So we can add it to the list of valid typos ➔ In future, we can also login with PWD_1

TypTop: Security Concern

Login: SpongeBob, PWD_2 = Patrick123Bank

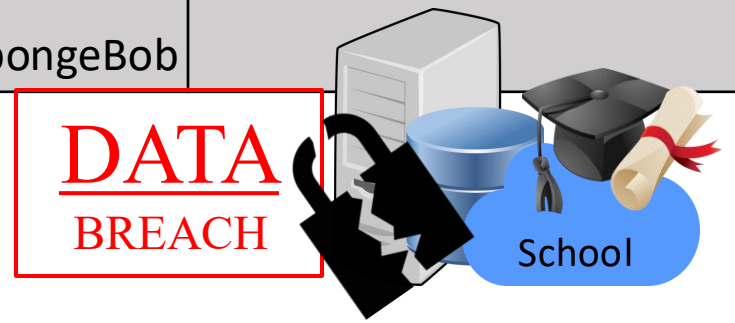
$c_{PWD_2} = \text{Enc}_{pk}(\text{Patrick123Bank})$

Cache	Incorrect Login Attempts (Encrypted)
$C_{sk,PWD}$ C_{sk,PWD_1} pk SpongeBob	c_{PWD_2}



Decrypt Waitlist using orig. PWD:

$\text{Patrick123Bank} = \text{Dec}_{sk}(c_{PWD_2})$



C_{PWD_2} didn't help me to find your school pwd, but I found it anyway.
BTW thanks for sharing your bank PWD!

Ideal World

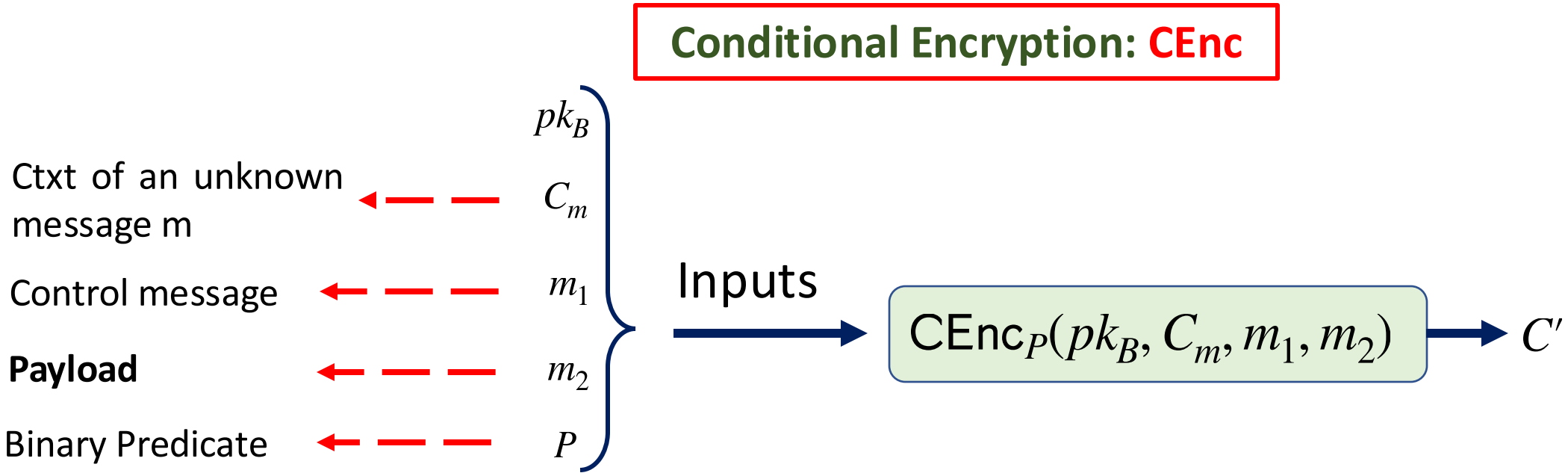
- What we want 🤔 : *Ideally*, we would like to ensure that only **plausible typos** are placed in the encrypted vault.
  - Plausible Typo: low hamming/edit distance, CAPSLOCK error

- Challenge 🥶 : We don't have a **plaintext copy** of the original password.

- Question 🤔 : How to decide if incorrect login attempt is plausible typo?

Our Solution: Conditional Encryption

- Extends public key encryption (KeyGen, Enc, Dec) with a ``*new conditional encryption*'' algorithm (CEnc).



E.g. Is Edit distance PWD_1 and PWD_2 is less than one? If Yes, then P(PWD_1, PWD_2) =1

Our Solution: Conditional Encryption

Properties:

If $P(m, m_1) = 1$ (messages are related):

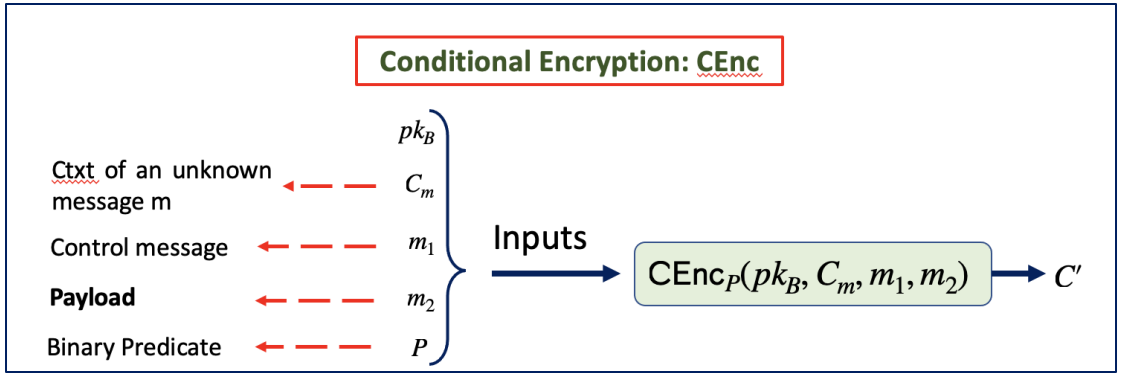
- then C' decrypts to payload m_2 (Correctness)

If $P(m, m_1) = 0$: C' reveals nothing about m, m_1, or m_2

- (Even if the attacker knows the secret decryption key sk!)

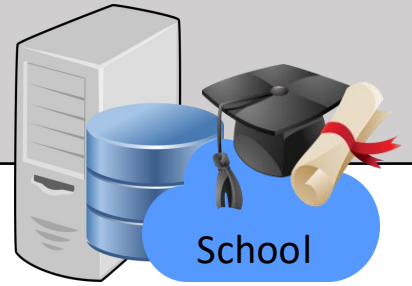
NOT satisfied by the regular/traditional PK encryption schemes:
➔ **New/Different security** notion

TypTop Application: Adversary may learn secret decryption key after a server breach.



TypTop with Conditional Encryption

Cache	Incorrect Login Attempts (Encrypted)
$C_{sk,PWD}$, C_{PWD}	$c_{PWD_2} = \mathsf{CEnc}_{pk}(pk, C_{PWD}, \text{Patrick123Bank})$
	$c_{PWD_3} = \mathsf{CEnc}_{pk}(pk, C_{PWD}, \text{pATRICK123SCHOOL})$
pk SpongeBob	



Decrypt Waitlist using orig. PWD = ``Patrick123school'':

$$P(PWD, PWD_2) = 0 \rightarrow \quad R = \mathsf{CDec}_{sk}(pk, C_{PWD_2})$$

$$P(PWD, PWD_3) = 1 \rightarrow \text{pATRICK123SCHOOL} = \mathsf{CDec}_{sk}(pk, C_{PWD_3})$$

Our Results

- We show how to construct:
  - CE for ***any*** predicate $P(\cdot)$ using circuit private FHE.
    - [+] General 🙂
    - [-] Computationally Expensive ☹️
  - ***Efficient*** CE for predicate tailored to password typos.
    - Small Hamming Distance, CAPSLOCK, Edit Distance One,
    - OR of above predicates.
- C++ Implementation of CE for password typo predicates
- Updated implementation of TypTop system using CE

Adobe Creative Cloud Query: Picture of Graduate Student writing C++ code

Building Conditional Encryption

- **Key Ingredients**
  - Pallier Cryptosystem (Partially Homomorphic)
  - Shamir Secret Sharing
  - Password Based Key Derivation Functions
  - Authenticated Encryption

Adobe Creative Cloud Query: Mixing material in pot over fire: the ingredients are labeled Shamir Secret Sharing, Pailier Encryption, Authenticated Encryption

Equality Predicate (Intuition)

- Conditional Encryption: $C = \mathrm{CEnc}_{pk}(C_m, m_1, m_2)$
- Using Paillier
  - ✓Addition Property: $\boxplus$ $\quad \mathrm{Enc}(m_1) * \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 + m_2)$
  - ✓Plaintext-Ctxt Multiplication: $\boxtimes$ $\quad m_1 * \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 . m_2)$

$$C = \mathrm{Pail}.\mathrm{Enc}_{PK}(m_2) \boxplus R \boxtimes \left[c_m \boxminus \mathrm{Pail}.\mathrm{Enc}_{K}(m_1) \right]$$

$$C = \mathrm{Pail}.\mathrm{Enc}_{PK}\left(m_2 + R\left(m - m_1\right) \right), R \in_R \mathbb{Z}_N$$

If $m = m_1$, then C decrypts to m_2
O.W., to random value $R' = m_2 + R(m - m_1)$

Caveat: Some technical details are being swept under the rug (see paper for full details).

CAPSLOCK via Equality Predicate

$$P_{\mathsf{CAPSLOCK}}(m_1, m_2) = 1 \textbf{ if } m_1 = \mathsf{InverCase}(m_2)$$

Conditional Encryption for Equality Predicate immediately yields a construction for CAPSLOCK Predicate

Hamming Distance Predicate (Intuition)

- **Key Idea:** Apply Conditional Encryption (for Equality Test) character by character
  - Control message for ith character is $m_1[i]$
  - Payload for ith character is ith secret share $[\![s]\!]_i$ of fresh symmetric key K
  - Recover $[\![s]\!]_i$ $\leftrightarrow$ we match at the ith character
  - Obtain enough shares to recover K iff Hamming Distance $\leq d$
    - n-d out of n secret sharing
  - Use symmetric key K to encrypt/decrypt payload

Caveat: Above construction is overly simplified and is not quite secure. See paper for the problem/fix.

Conditional Encryption for Relevant Predicates

- CAPSLOCK: use equality predicate
- Hamming Distance: see prior slide
- Edit Distance: Similar techniques (see paper)
- OR Composition: Concatenate ciphertexts from above predicates
- Typo Predicate: CAPSLOCK or Hamming Distance ≤ 2 or Edit Distance ≤ 1
  - [CWPCR:CCS17]: 78% of legitimate typos fit one of the above three categories

[CWPCR:CCS17] Chatterjee, et al. "The typtop system: Personalized typo-tolerant password checking." *ACM-CCS,* 2017.

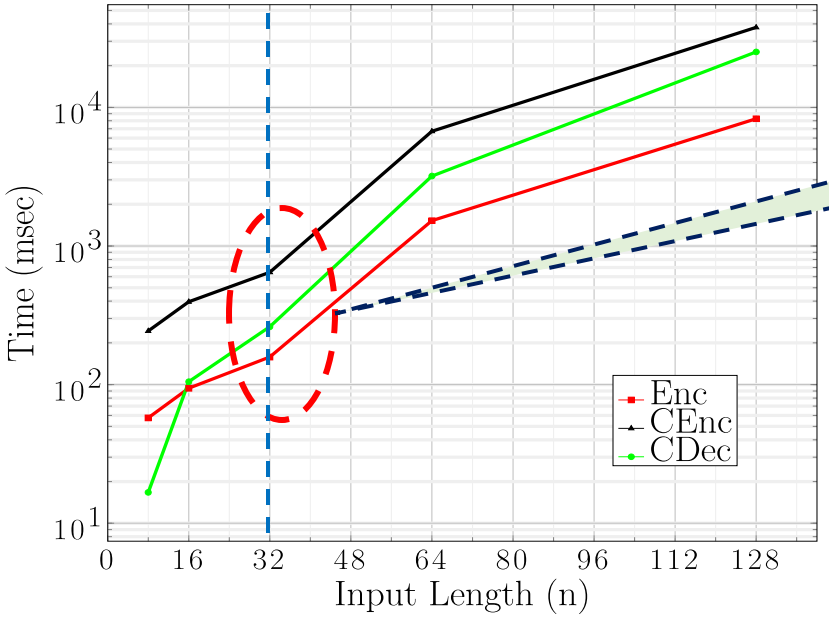
Implementation and Performance Evaluation

- C++ implementation of CE for the mentioned predicates
- C++ reimplementation of TypTop using CE (+Memory Hard Functions: Argon2id)
- On Github: https://github.com/mhassanameri/CondEncCCS24Artifact
- Certified by the ACM-CCS24 Artifact Evaluation Committee

Performance Evaluation (Typo predicate)

- CE for Typo predicate (n – pwd length):
  - ✓ Over 99.9% passwords have length ≤ 32.

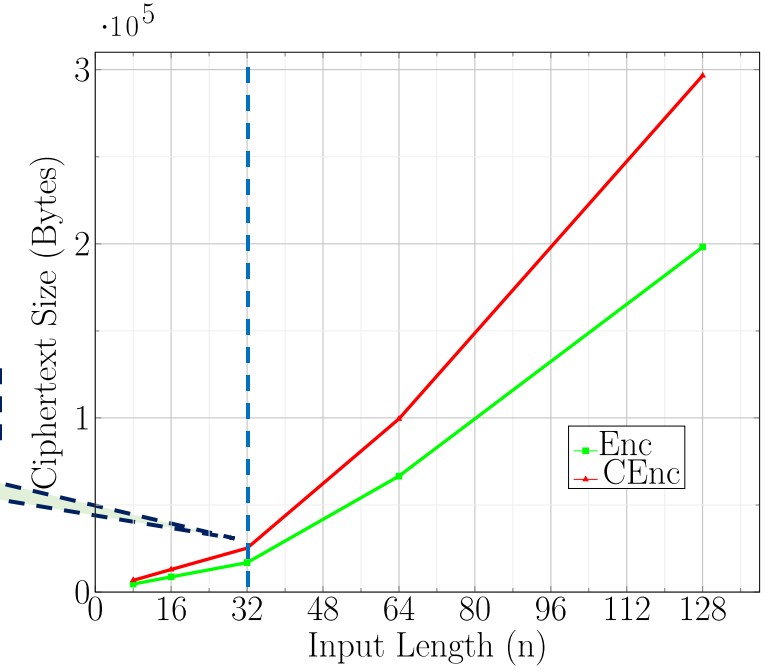
Reminder: Typo predicate ➔
``CAPSLOCK'' or ``Hamming Distance ≤ 2'' or ``Edit Distance ≤1''



(f) CEnc OR predicate

For n = 32 (chars):
time of all 3 Algs. < 1 (s)

For n = 32 (chars):
CTX size is < 25 KBytes



(g) CEnc OR predicate, CTX size

Thank You For Attention!