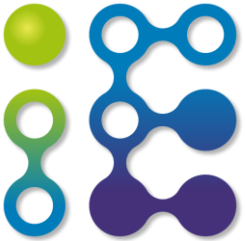
# *Omnes pro uno*
# Practical Multi-Writer Encrypted Database

*Jiafan Wang* and Sherman S. M. Chow

Department of Information Engineering
The Chinese University of Hong Kong, Hong Kong



# Storing/Collecting Data with Cloud

- Cloud services relieve the pressure of locally storing ever-growing data
- They also provide a platform for multi-writer applications, e.g.,
  - sensor network for crowdsourcing
  - machine learning
  - collective intelligence

- Yet, it is doubtful whether the cloud server can be trusted
- *How to secure outsourced data while enabling efficient retrieval?*

# Searching Encrypted Data

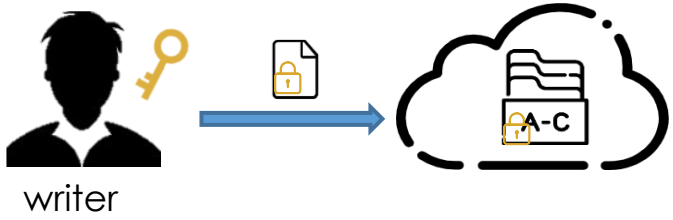
- Standard encryption prevents any data retrieval/search
- Multi-client ORAM [**C**FLM20] or FHE are too heavyweight
- *Searchable encryption (SE)* is promising
  - realize nice security-efficiency tradeoff for searching encrypted data
- Two well-studied SE notions
  - Dynamic Searchable Symmetric Encryption (DSSE) [KPR12]
  - Public-Key Searchable Encryption (PKSE) [BDOP04]

- *Do they perfectly fit cloud-based multi-writer applications?*

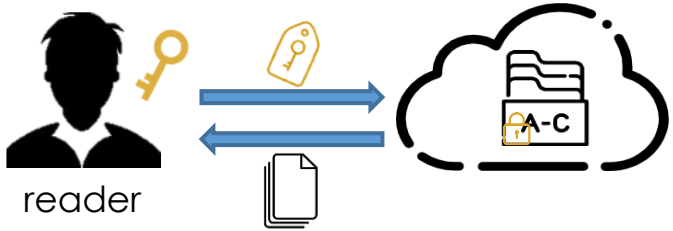


# Representative SE Notions: DSSE

- Update/Write




- Search/Read

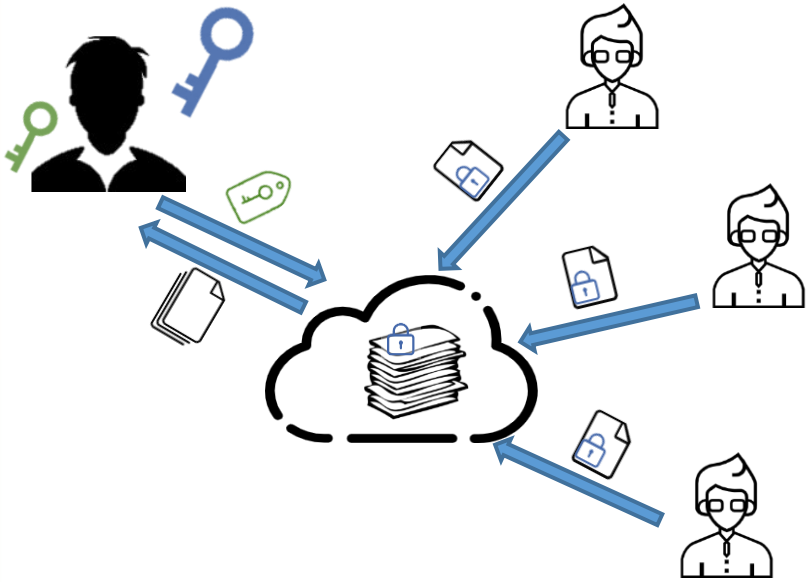



✓ Sublinear search
- index data for efficiency

✓ Forward privacy
- keep privacy of future updates
- by "changing" update tokens
- to invalidate old search tokens

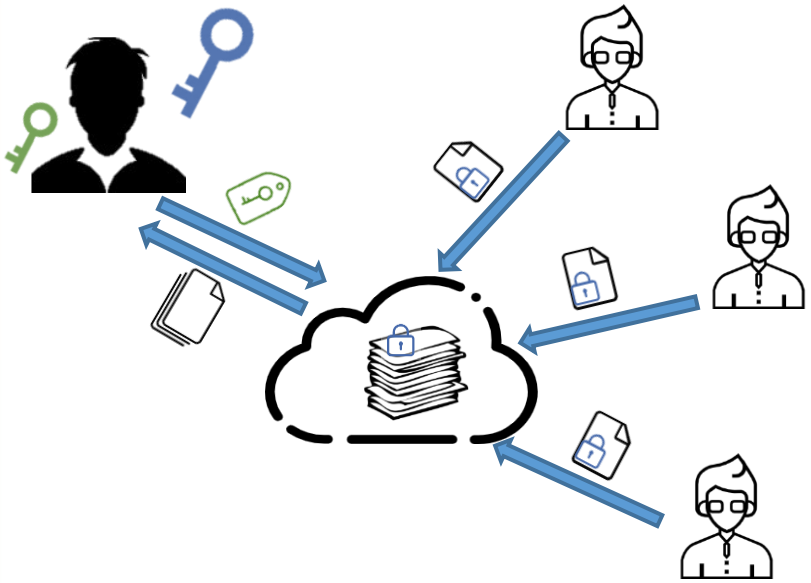
✗ No multi-writer support
- writer = reader = secret-key owner
- fails for contributive applications

# Representative SE Notions: PKSE



- ✓ Multiple writers for a reader
  - no secret-key distribution
  - no synchronous communication
- ✘ Often require linear testing
  - challenging to jointly index with no shared secret/coordination
- ✘ Hard to be forward private
  - writers don't know any search state
    - nor when to "change" update tokens

# Few Developments of PKSE



- Forward-private PKSE [ZQCZ19], like PKSE, requires linear testing
- SPCHS [XWW+15] (not forward private) features sublinear search
  - a pairing for each retrieval step
- No prior multi-writer solution achieves forward privacy and sublinear search simultaneously

# Towards the Best of Both worlds

**DSSE**

✓ Sublinear search

✓ Forward privacy

✘ No multi-writer support

**PKSE**

✓ Multiple writers for a reader

✘ Often require a linear scan

✘ Hard to be forward private


- *How to simultaneously realize nice features of PKSE and DSSE?*

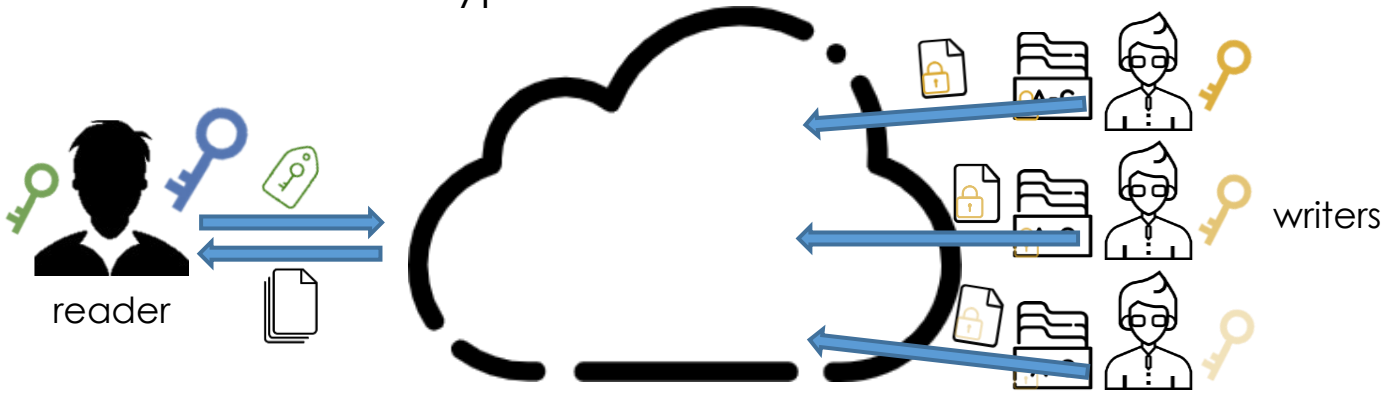
**New Notion – Hybrid Searchable Encryption (HSE)**

# HSE Model

- A *reader* holds public/master secret key pair
- Multiple *writers*, each create a DSSE instance
- A *server* stores encrypted databases (with all DSSE instances)

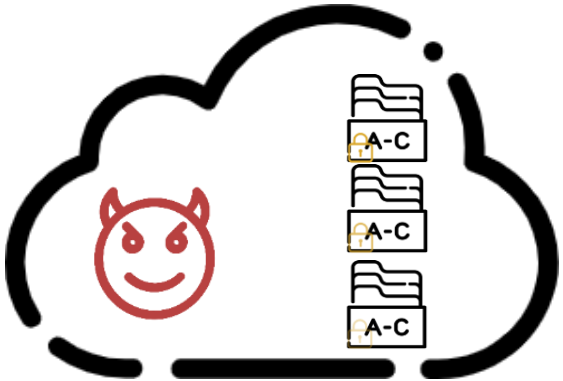
➢Each writer independently updates its own DSSE

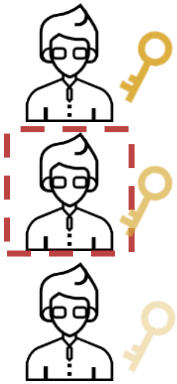
➢The reader searches encrypted databases with master secret



# HSE Security

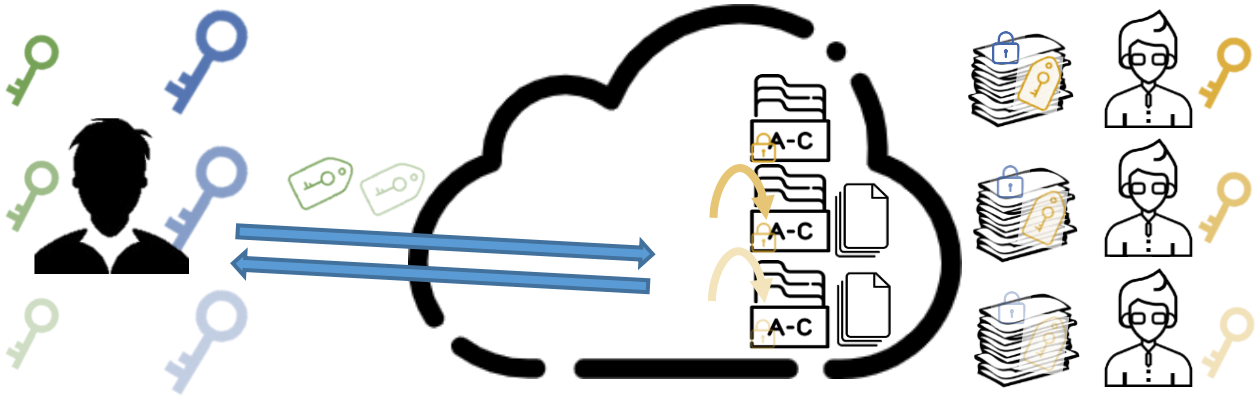
- Against semi-honest server and *potentially corrupted writers*
- Parameterized by leakage functions
  - capture leakage during Setup, Search, Update, and Corruption
- Typically, no *leakage analysis* for PKSE; no *corruption* in DSSE



# Generic Construction G-HSE

- G-HSE: DSSE + PKSE/Anonymous Identity-based Encryption (id = keyword)
  - The reader initializes $n$ IBE instances for $n$ writers
  - Each writer encrypts DSSE search tokens with corresponding IBE
    - for the 1st regular DSSE update of a keyword, "1st status" is recorded by a bit list
  - A search on targeted writers gets their DSSE tokens via IBE decryption



# Nice Features of G-HSE

- Multi-writer support without getting secrets from the reader
  - only need reader public key to encrypt, only cloud server is online
  - each writer *independently* updates, no synchronization with reader

- Search Complexity O(|W| + $a_w$)
  - W: active keywords from target / interested writers (IBE decryption)
  - Multi-writer updates need public-key operations to differentiate
  - $a_w$: # of updates on target keyword $w$ (*symmetric-key* DSSE operation)
  - Typically, sublinear in the database size



# Shortcomings of G-HSE

- Reader can confine search scope to specific (targeted) writers
  - new for multi-writer SE, benefit efficiency and security
  - yet, search token size is linear in # of target writers (or writer classes)
- No forward privacy (search tokens work forever like PKSE)
  - underlying DSSE search token is encrypted *once for each keyword*
  - i.e., "statically generated" independent of whether search happened
- We propose two crypto primitives to resolve each concern



# ID-Coupling Key-Aggregate Enc.

- Key-Aggregate Encryption [C**C**T+14] is done w.r.t. a *class*
- KAE can aggregate decryption abilities of *any* subset of classes
- Secret key/Ciphertext size is O(1), independent of # of classes

- ICKAE couples such an aggregate key with an identity
  - When being used for PKSE: identity, *i.e.*, keyword, is anonymized
  - When being used for HSE: each writer belongs to a *class*

- Our ICKAE scheme helps to upgrade HSE
  - O(1) token size and reader storage
  - Efficient construction (only 2 pairings)

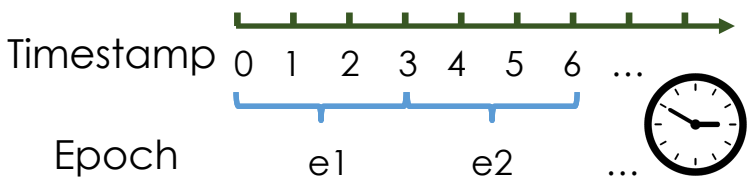
# (Traditional) Forward Privacy

- Search token can't work for *any* future updates [Bost16, L**C**17]
- A usual trick: *after* each search, reader *refreshes* database/state

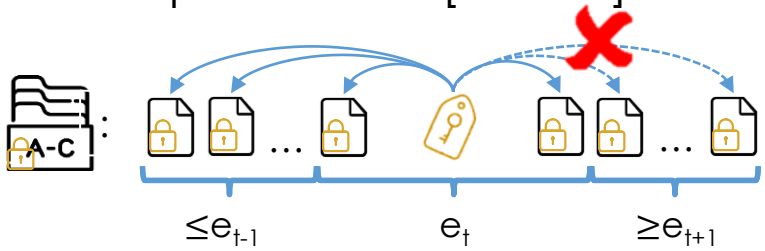

- *Inapplicable* to (multi-writer) HSE
  - e.g., writers have *no idea* whether the reader has recently searched for the keyword they want to update

# Epoch-Based Forward Privacy

- Assume publicly-known epoch info.
  - i.e., a subset of system timestamps
  - from a (loosely synchronized) clock

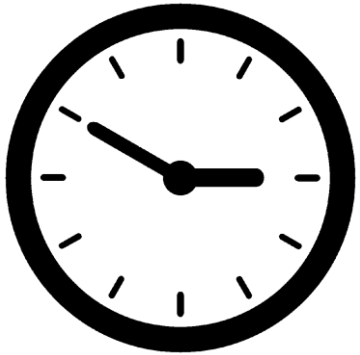



- A search token can't work for updates *at future epochs*
  - inspired by forward-private PKE [CHK03] and PKSE [ZQCZ19]



# Epoch-Based Forward-Private DSSE

- Time as an indicator for refreshment
  - weaker yet compatible with (multi-writer) HSE



- Our E-DSSE scheme achieves a new tradeoff: *forward privacy* vs. *temporary delegations*
  - *a shorter* epoch means *higher* forward privacy
  - but it trades flexibility, issued tokens become invalid sooner

# Forward-Private HSE

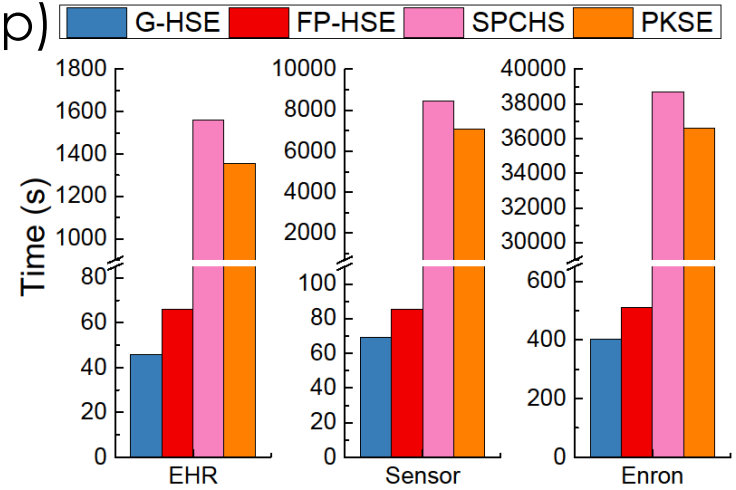
- *FP-HSE* instantiates G-HSE with ICKAE and E-DSSE
  - Inherits O(1)-size token and reader storage from ICKAE
    - independent of the epoch length or # of target writers for confined search
  - Retains forward privacy from E-DSSE (in a multi-writer setting)
    - use epoch information as an indicator for refreshment

- It uses a *rebuild trick* to remove ciphertexts of old epochs
  - Writers could help *rebuild* ICKAE-encrypted tokens at a new epoch
  - to maintain sublinear search time with an acceptable rebuilding time



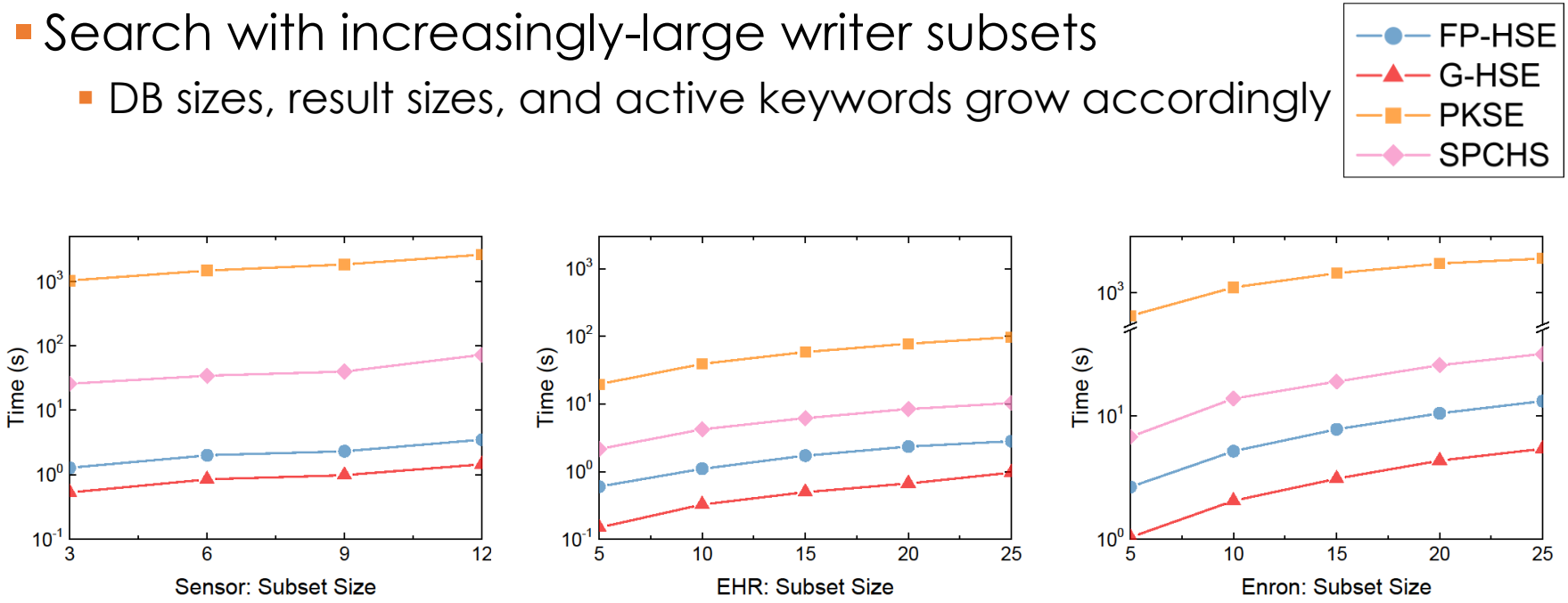
# Experimental Evaluation

- Dataset
  - EHR of diabetes patient: 101K records from 130 hospitals
  - Sensor data of room climate: 540K records from 12 sensors
  - Enron e-mails: 510K e-mails from 147 employees
- Building time of FP-HSE (and speedup)
  - EHR ~ 66s (23× SPCHS, 20× PKSE)
  - Sensor ~ 85s (98× SPCHS, 82× PKSE)
  - Enron ~510s (75× SPCHS, 71× PKSE)
- Average rebuild time per writer
  - EHR ~395ms
  - Sensor ~505ms
  - Enron ~1.79s



# Experimental Evaluation

- Search with increasingly-large writer subsets
  - DB sizes, result sizes, and active keywords grow accordingly



# Summary

Contact: {wj016, sherman}@ie.cuhk.edu.hk

- Hybrid Searchable Encryption – a new notion uniting PKSE and DSSE
  - formalization of syntax, security, and desirable properties
  - generic construction featuring multi-writer support and sublinear search
- Towards *HSE with forward privacy and compact token*, we propose
  - identity-coupling key-aggregate encryption (ICKAE)
    - enrich "cloud cryptography" for access control with search
  - epoch-based forward-private DSSE (E-DSSE)
    - a new formulation with a new tunable treatment of forward privacy
  - (our building blocks are of independent interests)
- We hope to see more research / deployment of *HSE*