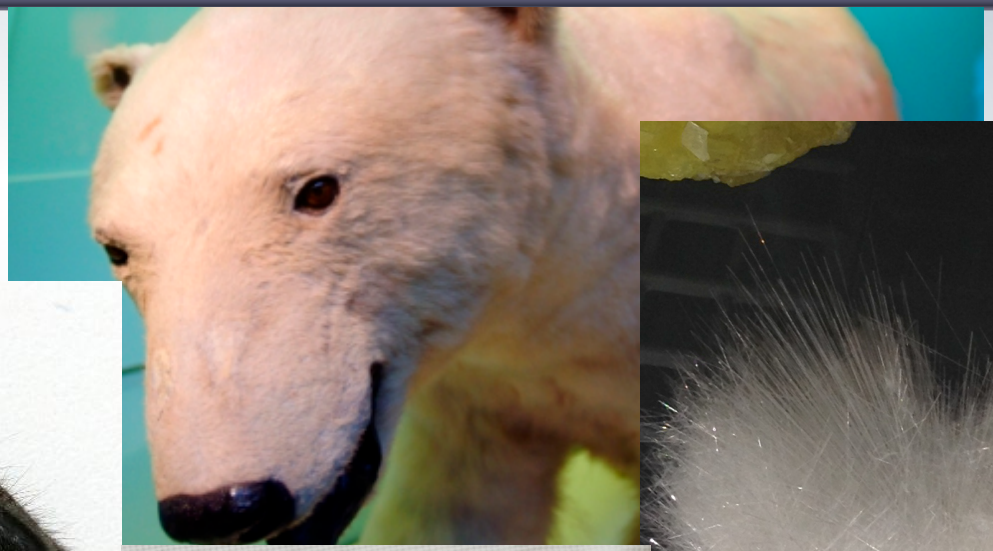
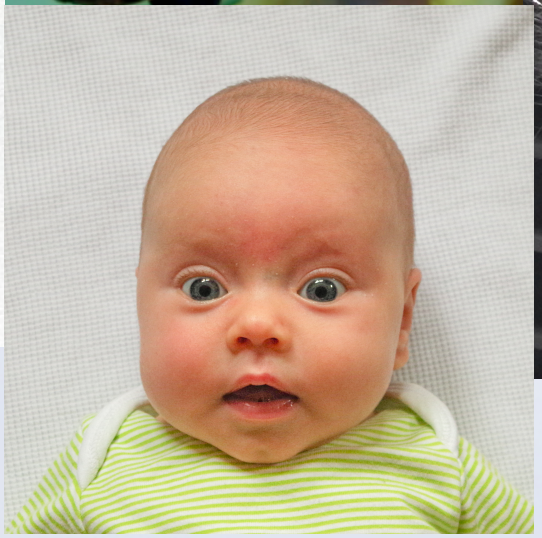
Efficient Fuzzy Search on Encrypted Data

Alexandra Boldyreva, Georgia Tech
Nathan Chenette, Clemson University

Fast Software Encryption 2014
London, UK

Fuzziness in NHM?



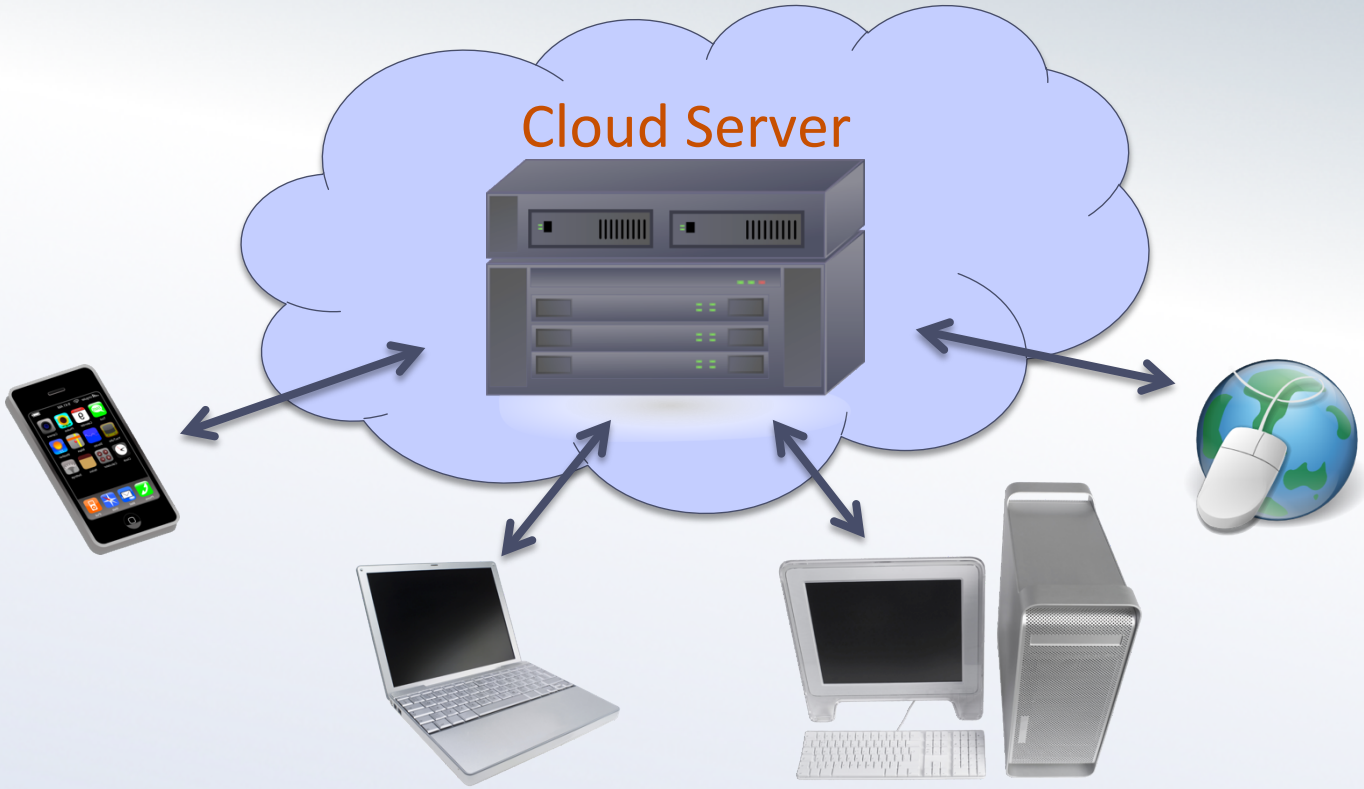


Outline

- Background and motivation for efficient search on encrypted data
- Efficient Fuzzy-Searchable Encryption (EFSE) for efficient error-tolerant (fuzzy) queries on encrypted data
- Primitives and optimal EFSE security
- General "tag-encoding" construction template and security conditions
- Optimally-secure scheme
- More space-efficient, less secure schemes

Background and Motivation

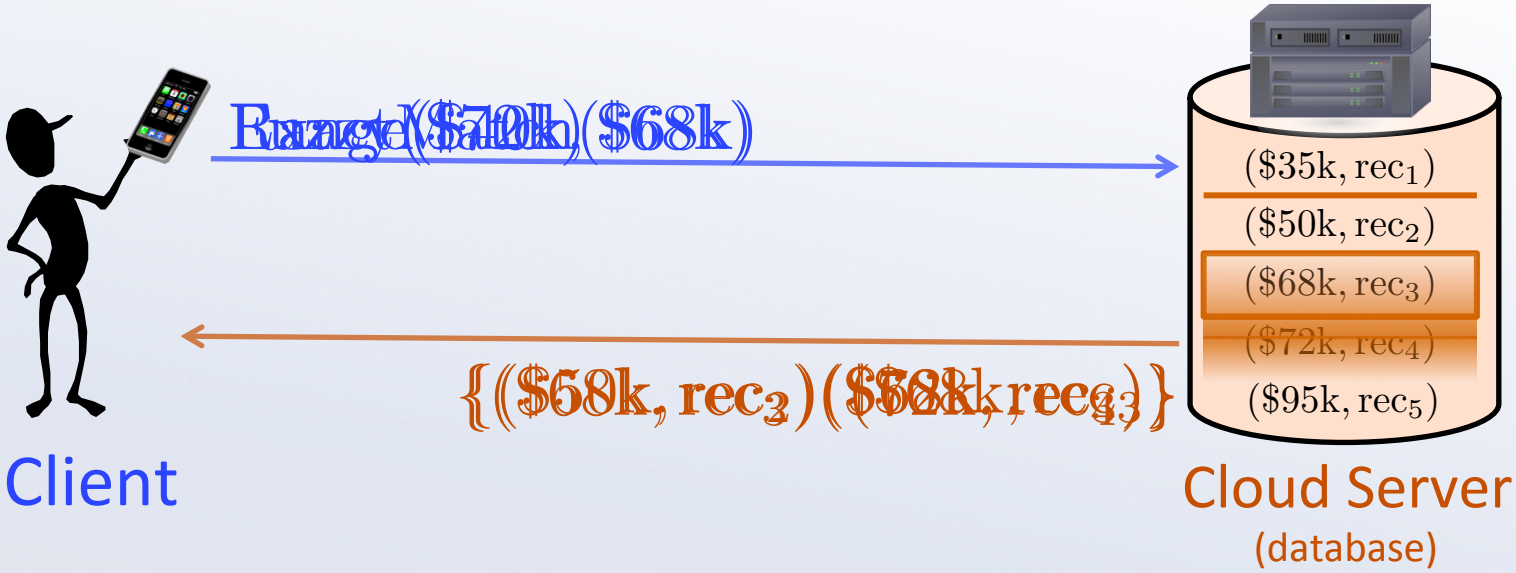
Cloud Storage



- Advantages: mobility, flexibility, decentralization, division of labor, lower costs
- Major disadvantage: insecurity

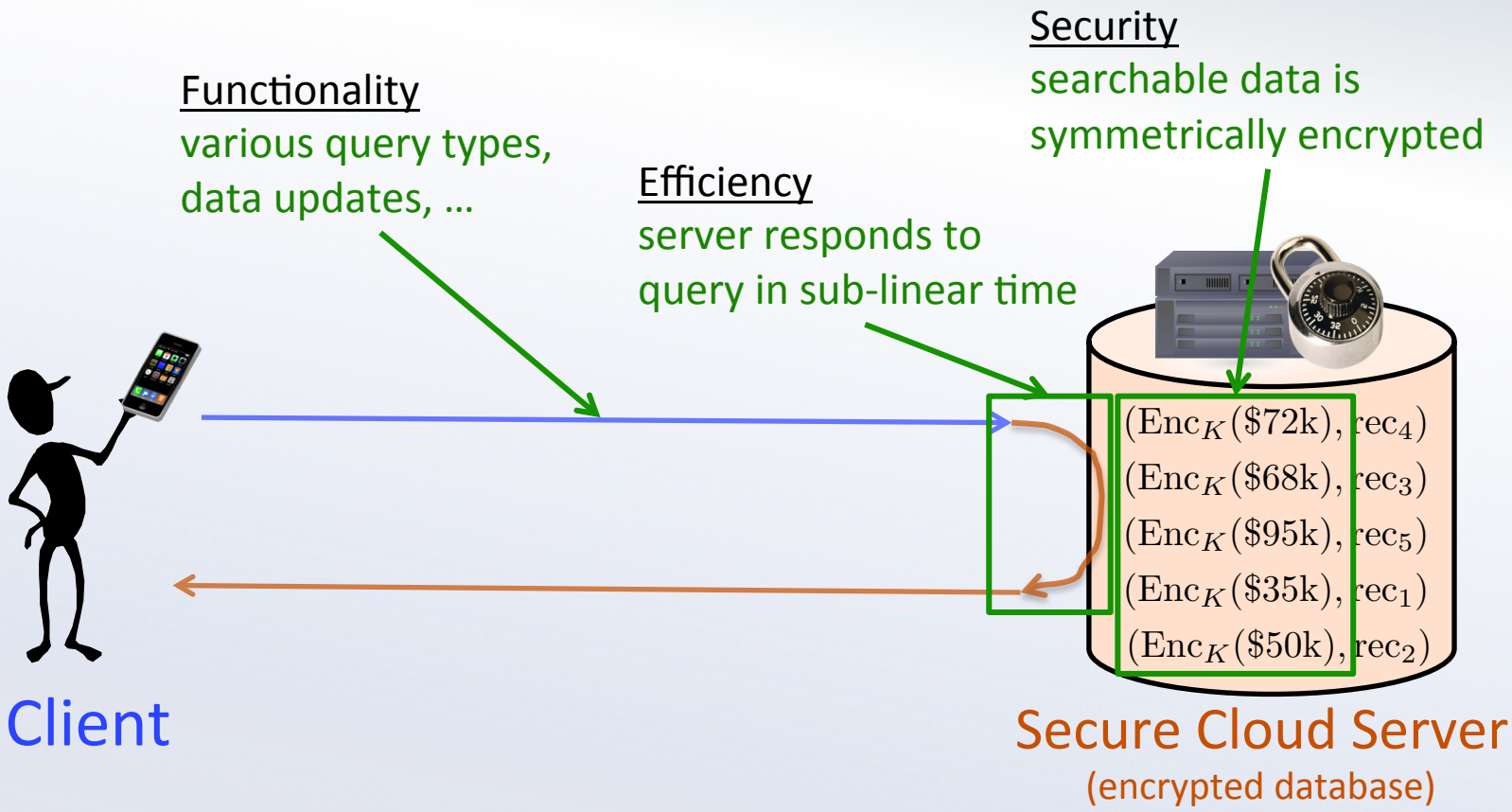
Cloud Storage

- A.k.a. Database-as-a-Service
- Server efficiently responds to client's queries/updates
  - Query efficiency: search time sub-linear in database size
  - Query functionality: exact-match, range, error-tolerant (fuzzy),…



Secure Cloud Storage: Goals

- Three goals: security, efficiency, functionality



Efficient Searchable Encryption

- Efficiency, security, and functionality are at odds
  - E.g., strong encryption requires linear search time

- The study of schemes balancing these goals is efficient searchable encryption (ESE)
  - Cryptographic efforts often focus on strong security
  - Practitioners wonder: how much security is possible without sacrificing efficient functionality?

Past Results in Searchable Symmetric Encryption (SSE)

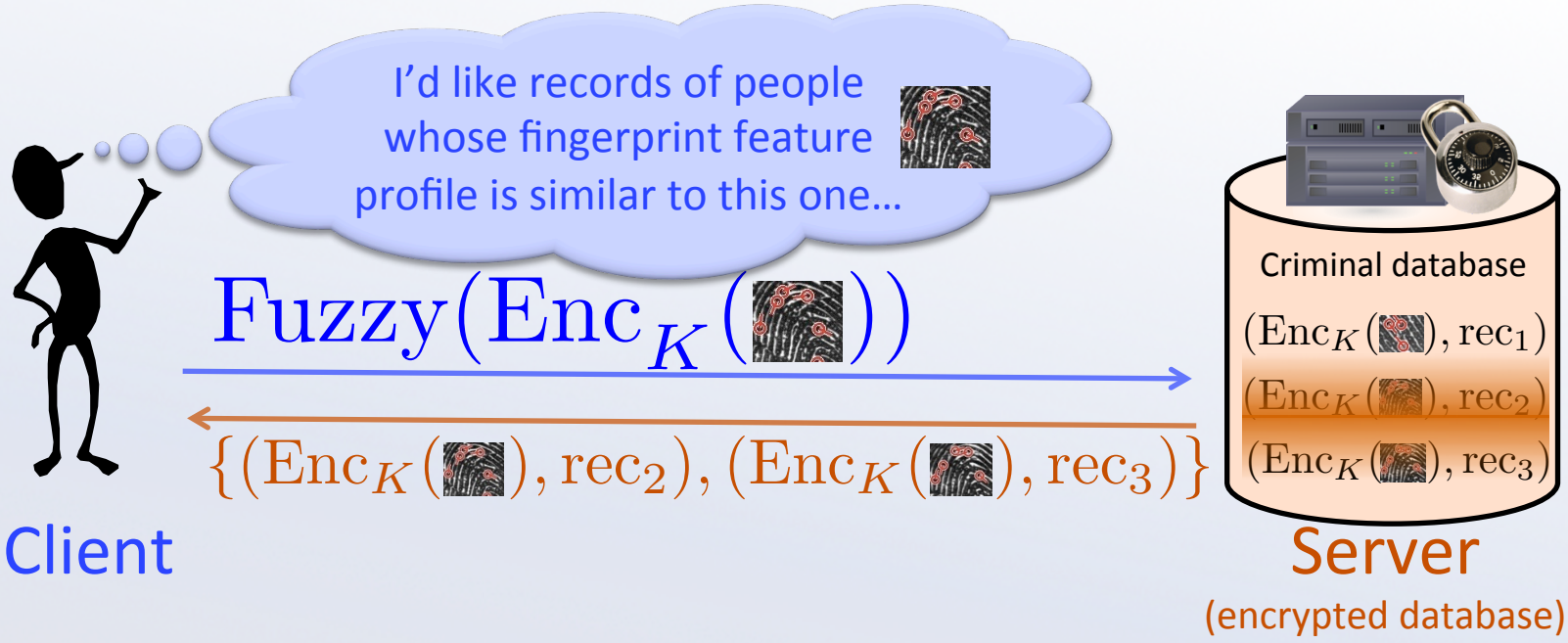
	Security	Efficiency	Functionality
Fully Homomorphic Encryption [RAD78,G09] Oblivious RAM [GO96]	Semantic+	Impractical	All query types
Exact-match SSE [SWP00,G03,GSW04,CM05] Range-query SSE [BW07, SBCSP07]	Semantic+	Linear+	Exact-match Range
Exact-match ESE via static indexes[CGKO06,SvLDHJ10,KO12] Similarity ESE via static indices [KIK12]	Adaptive semantic	Sub-linear	Exact-match Fuzzy Limited dynamic data updates
Ad-hoc order-preserving encryption [AKSX04] Ad-hoc efficient fuzzy-searchable encryption [LWWCRL10]	Undefined/unknown	Sub-linear	Range Fuzzy
Efficiently-searchable authenticated encryption [ABO07]	Leaks only equality	Sub-linear	Exact-match
Order-preserving encryption [BCLO09,BCO11]	Pseudorandom OP, Low-order-bit 1way	Sub-linear	Range
Efficient fuzzy-searchable encryption [BC14]	Leaks only closeness and equality*	Sub-linear*	Fuzzy

Goal

- Past fuzzy-searchable encryption schemes
  - [KIK12] scheme relies on knowing the data in full in advance (no dynamic updates)
  - [LWWCRL10] scheme is ad-hoc and has no formal security analysis (we show that it has some security limitations)
- Our goal: provide the first *provably-secure* solutions for supporting efficient fuzzy search on *dynamically-updatable*, symmetrically encrypted data

EFSE and Motivation

- Intuitively, efficient fuzzy-searchable encryption (EFSE) refers to schemes where fuzzy queries can process in the ciphertext domain
- Useful when data is inherently approximate or error-prone (e.g., biometric data)



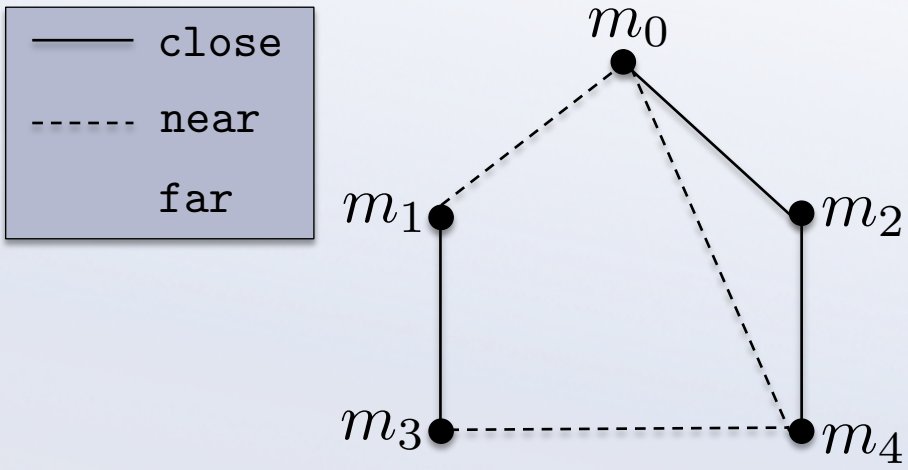
EFSE Primitives and Security Notion

Closeness

- How to define "closeness" of messages (that we want ciphertexts to reveal)?
- Closeness domain: domain $\mathcal{D}$ and closeness function Cl

$$\mathrm{Cl} : \binom{\mathcal{D}}{2} \to \{\texttt{close}, \texttt{near}, \texttt{far}\}$$

- Useful to characterize a closeness domain graph-theoretically

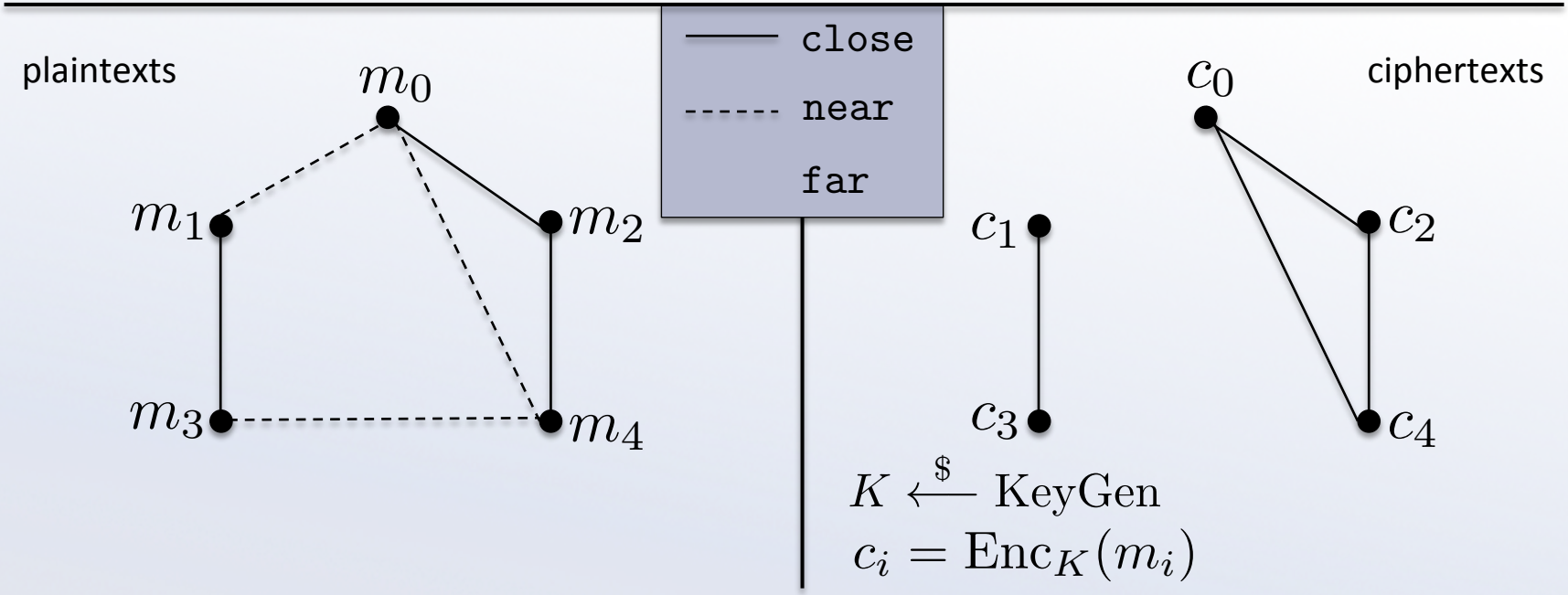


Closeness graph:
`close` edges

Nearness graph:
`near` and `close` edges

Encryption Leaking Closeness

- Essentially, a symmetric encryption scheme from one closeness domain to another is fuzzy-searchable (FSE) if encryption sends close messages to "close ciphertexts" and far messages to "far ciphertexts". We also require FSE schemes to leak equality.



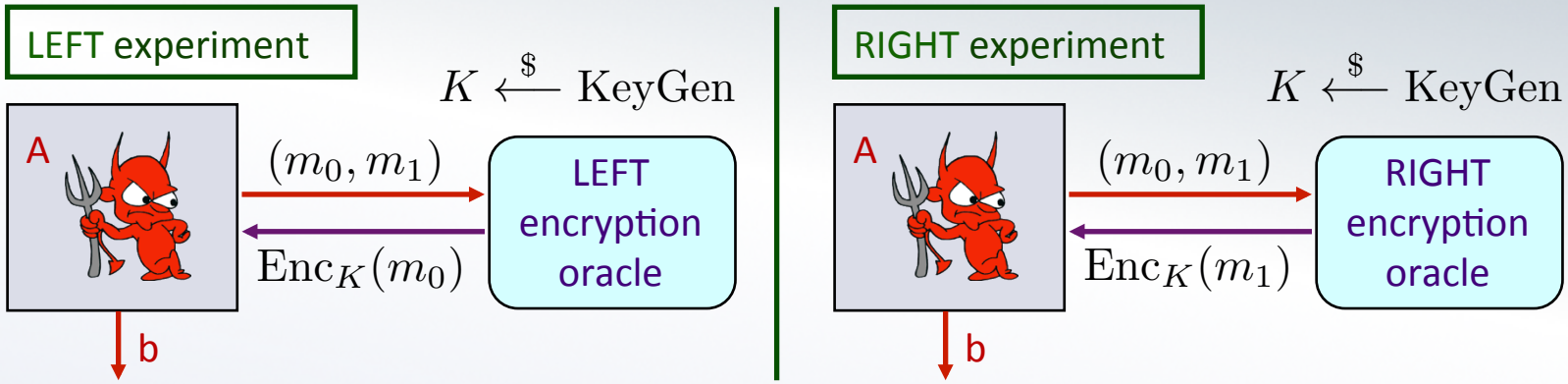
Efficient Fuzzy-Searchable Encryption

- To be *efficient* fuzzy-searchable (EFSE), an FSE scheme must enable finding close ciphertexts to a given ciphertext *efficiently* (sub-linear)

Optimal Security for FSE

- Optimally, an FSE scheme will leak only what it is supposed to: equality and closeness of messages
- We weaken IND-CPA to IND-CLS-CPA: indistinguishability under same-closeness-pattern chosen-plaintext attacks

IND-CLS-CPA-Security



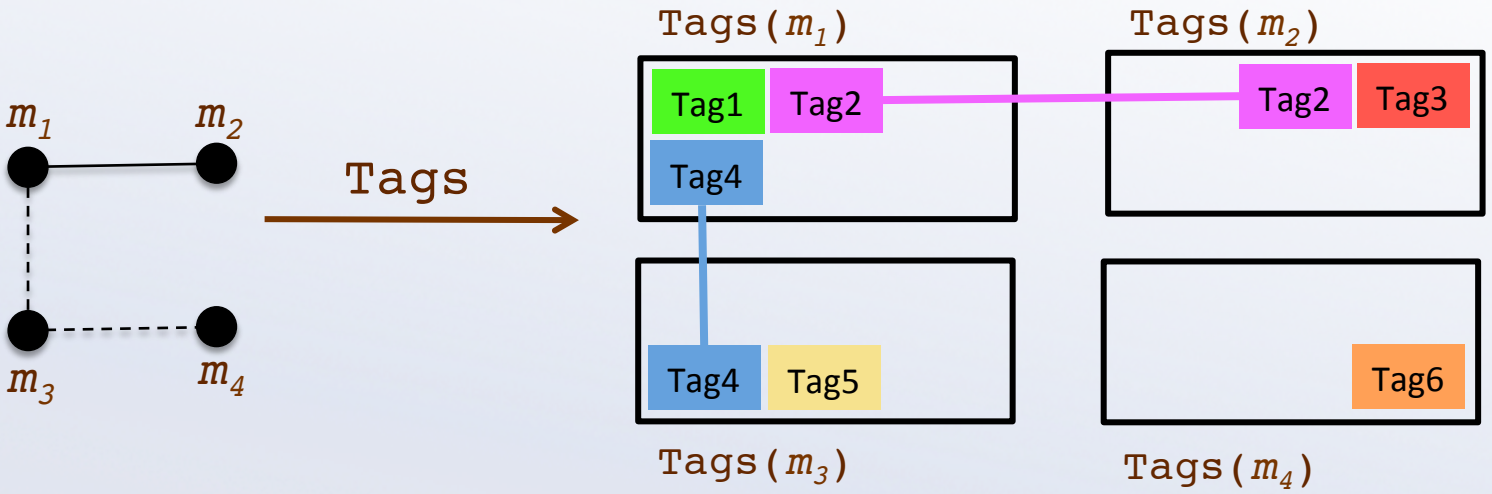
- Restriction: Left-right queries (m_0, m_1) must have the same equality and closeness pattern
  - That is, $m_0^i = m_0^j$ if and only if $m_1^i = m_1^j$

    and otherwise $\mathrm{Cl}_{\mathcal{D}}(m_0^i, m_0^j) = \mathrm{Cl}_{\mathcal{D}}(m_1^i, m_1^j) \quad \forall i, j$
- We call an OPE scheme IND-CLS-CPA-secure if no efficient adversary can output 1 with noticeably different probabilities between the two experiments.

General Tag-Encoding Construction

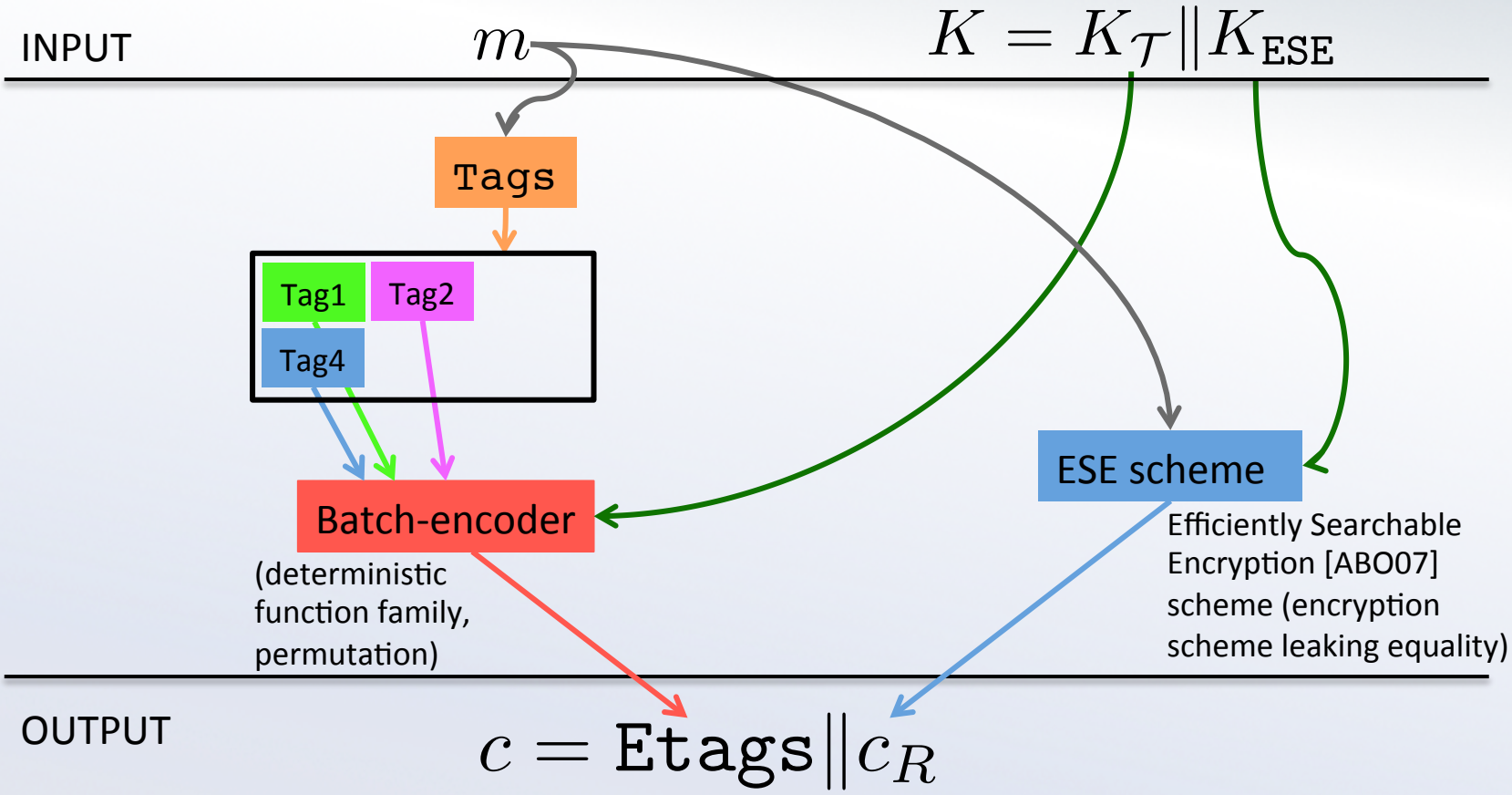
Closeness-Preserving Tagging

- A closeness-preserving tagging function (CPTF) is a function `Tags` from messages to sets of "tags" such that
  - tag sets of close message pairs intersect
  - tag sets of far message pairs are disjoint
  - (tag sets of near message pairs are unrestricted)



Tag-Encoding Template Construction

- Template encryption given a tagging function $\mathtt{Tags}$:



Using the Construction for Fuzzy Search on Encrypted Data

$$c = \texttt{Etags} \| c_R$$

- The encoded-tags leak closeness
  - Close ciphertexts overlap in `Etags`
  - Far ciphertexts have disjoint `Etags`
  - To implement efficient fuzzy search, maintain (say) a search tree indexed by encoded-tags
- The ESE output, c_R, leaks equality

Correctness and Security Conditions

	Tags	Batch-encoder	ESE scheme [ABO07]
Conditions for EFSE correctness	is a CPTF with small max-number-of-tags over the message space	is collision-free	
Conditions for optimal IND-CLS-CPA-security	is "consistent"	is PP-CBA (privacy-preserving under chosen batch attacks)	is IND-DCPA [BKN04] (indist. under distinct chosen-plaintext attacks)
Recommended instantiation	[see specific constructions]	Blockcipher-based pseudorandom permutation	Blockcipher-based [ABO07]
Condition for IND-CLS-CPA-insecurity	is not "consistent"		
	??	✓	✓

Consistency of a CPTF

- A CPTF $\mathtt{Tags}$ is consistent if *any* two message sets $\{m_0^1, \ldots, m_0^q\}$ and $\{m_1^1, \ldots, m_1^q\}$ having the same equality and closeness pattern overlap in the same number of tags, i.e.,

$$\left| \bigcap_{i \in [q]} \mathtt{Tags}(m_0^i) \right| = \left| \bigcap_{i \in [q]} \mathtt{Tags}(m_1^i) \right|$$

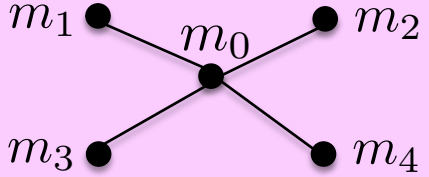
- **Theorem**. Consistency of $\mathtt{Tags}$ is necessary and sufficient (given the other conditions) for IND-CLS-CPA-security of the tag-encoding construction

Specific Constructions

Optimally-Secure Construction

- Let $G = (V, E)$ be the closeness graph after possibly adding dummy messages and edges to make vertex degree uniform
- Define $\texttt{TagEdges}(m) = \{e \in E \mid m \in e\}$

Ex:



$\texttt{TagEdges}(m_0) = \{m_0m_1, m_0m_2, m_0m_3, m_0m_4\}$

  - CPTF: close messages share an edge, far messages do not
  - consistent: number of edges shared by two isomorphic message-sets is equal
- Thus, the associated scheme is IND-CLS-CPA-secure as long as the max message degree is small

IND-CLS-CPA-Insecurity of Past Scheme

- This is an improvement over the previous EFSE scheme from [LWWCRL10], which is not IND-CLS-CPA-secure
- Its basic idea is to tag neighbors in the closeness graph, and fits into our tag-encoding template with CPTF $\texttt{TagNbs}(m) = \{m' \in V \mid \{m, m'\} \in e\} \cup \{m\}$

Ex:

$\texttt{TagNbs}(m_0) = \{m_0, m_1, m_2, m_3, m_4\}$

- This CPTF is not consistent, so the scheme is IND-CLS-CPA-insecure

Unavoidable Space-Inefficiency

- Both (secure) edge-tagging and (insecure) neighbor-tagging schemes are often space-inefficient
  - Ciphertext size linear in max closeness degree
- However, we show this ciphertext length is necessary in order to support EFSE on arbitrary closeness domains
  - Smaller ciphertexts cannot hold enough information to precisely describe closeness relationships in an arbitrary domain

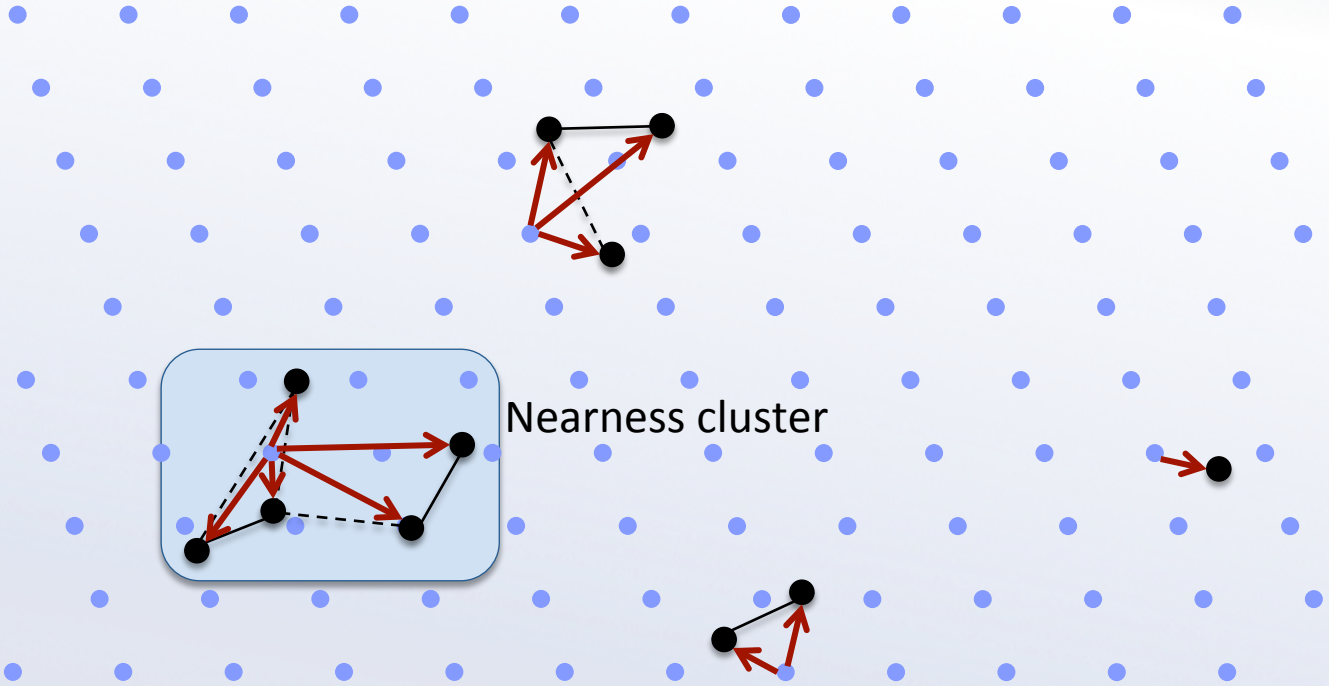
Space-Efficient EFSE

Relaxing Requirements

- The impossibility result relies on a very strict notion of closeness
- Can we improve space-efficiency for EFSE on closeness domains with "nearness"?
  - Recall: near messages can be sent to close or far ciphertexts
  - Unfortunately, having "more nearness" does not seem to improve space-efficiency if aiming for IND-CLS-CPA
- We need new notions of security to evaluate such schemes
- We focus on practical closeness domains: real multi-dimensional spaces with closeness defined by a metric and (close and near) thresholds

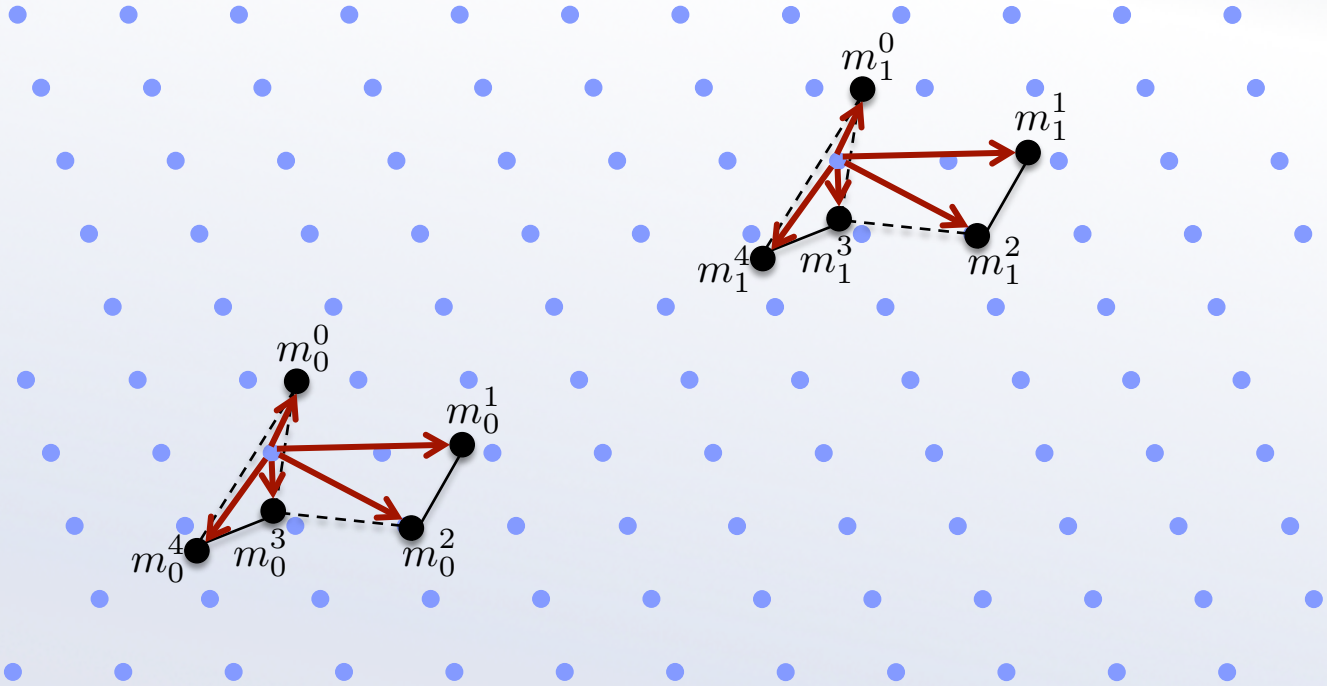
Macrostructure Security

- Defined with respect to a regular lattice $\mathcal{L}$ in $\mathbb{R}^n$
- Intuitively, hides all information except message location modulo the lattice for each nearness cluster



Macrostructure Security

- The notion requires that nearness clusters with same message locations modulo the lattice have indistinguishable ciphertexts

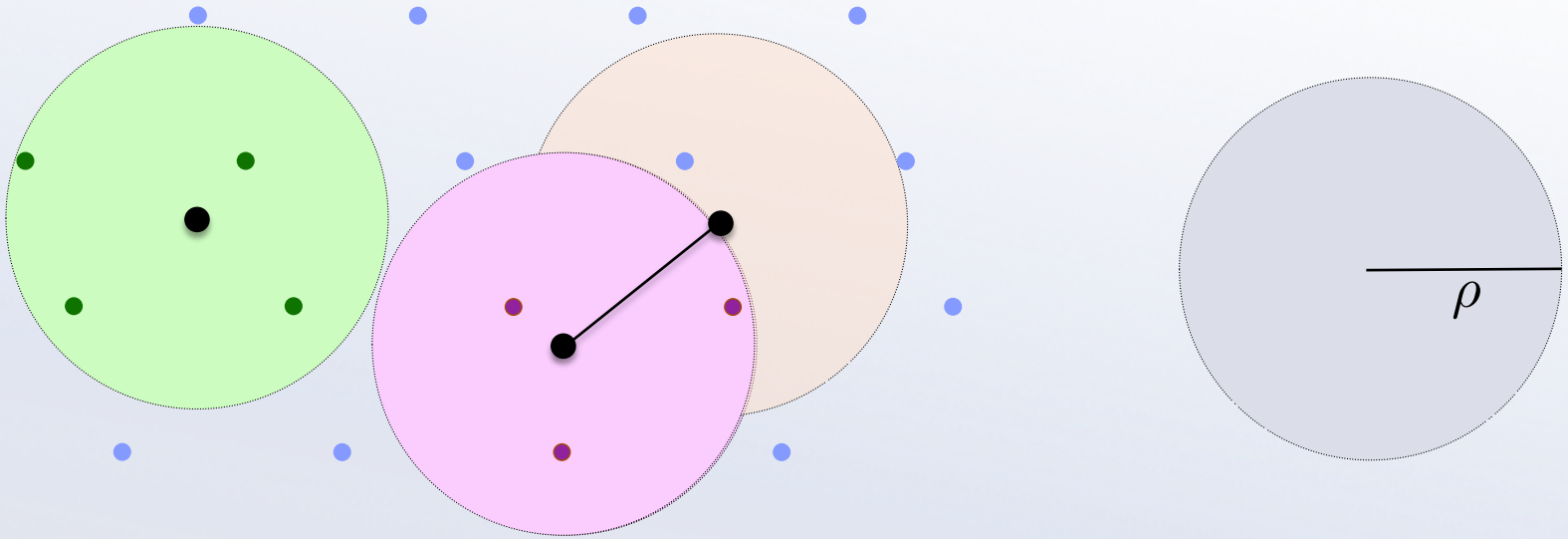


Macrostructure Security

- Relationships within a nearness cluster may be totally leaked, but only "small-bit" information is leaked about disconnected messages
  - Useful in applications where this leakage is acceptable
- We call this macrostructure security with respect to the lattice (MacroStruct-$\mathcal{L}$-secure)
- On $\mathbb{R}^n$ with closeness defined by a metric and threshold, attainable through a general construction given a valid anchor radius for $\mathcal{L}$

Anchor Radius Construction

- Consider balls of radius ρ centered at each message. ρ is a valid anchor radius if
  - Close message pairs' balls always contain a common lattice point
  - Far message pairs' balls never share a lattice point
- Lattice points within ρ of a message are its anchor points



Anchor Radius Construction

- Construction: use tag-encoding template with tagging function sending a message to its anchor points
- Results in a macrostructure-secure scheme
- We propose possible lattices and anchor radii for various dimensions, and discuss their domain-flexibility and space-efficiency

Conclusion

Conclusion

- Foundational cryptographic study of EFSE
- Primitives, appropriate security notions, and the first provably-secure EFSE schemes
  - Closeness domain, EFSE, tag-encoding template
  - Optimally-secure scheme
    - Space-inefficiency is unavoidable for the application
  - More space-efficient schemes that meet a natural new security notion and may be useful for applications such as in secure cloud storage

Thanks!