ComputerScience

# GRECS: Graph Encryption for Approximate Shortest Distance Queries

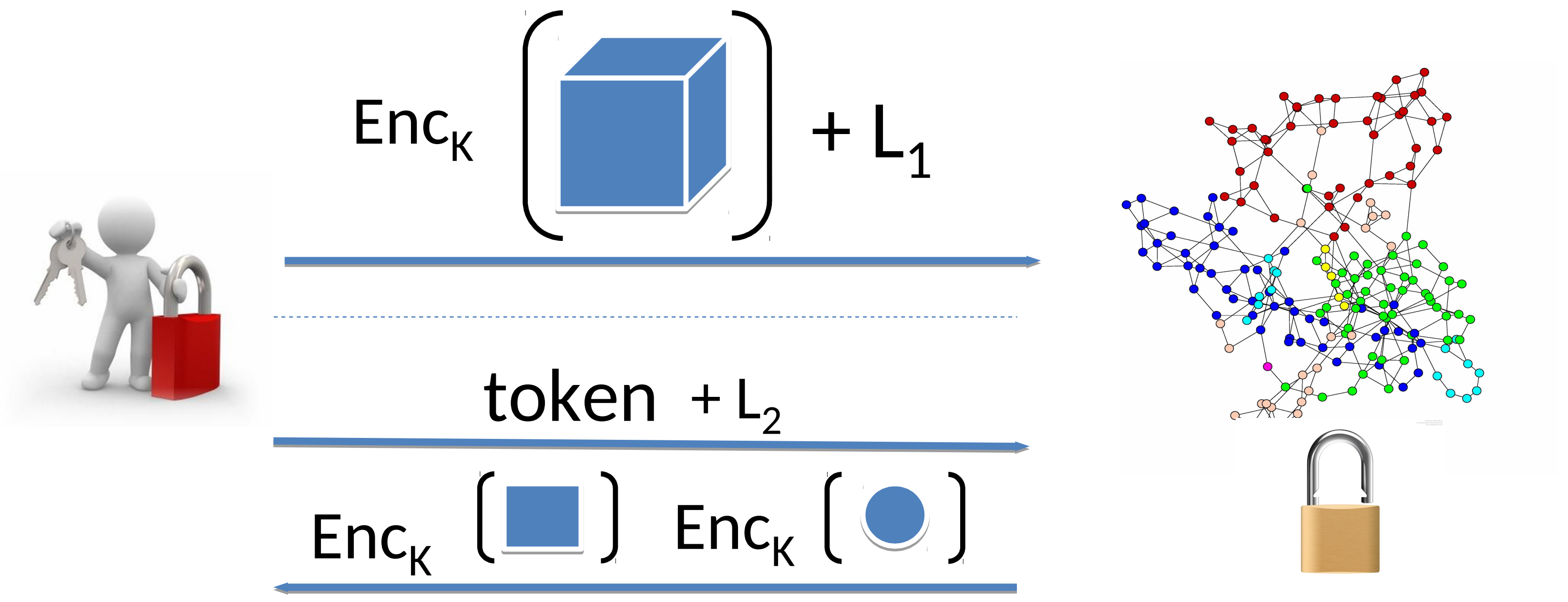
Boston University

Xianrui Meng┼, Seny Kamara˥, Kobbi Nissim ¶, George Kollios ┼

┼ Boston University, ˥ Microsoft Research, ¶ Ben-Gurion University/Harvard University

## ABSTRACT

We propose graph encryption schemes that efficiently support approximate shortest distance queries on large-scale encrypted graphs. Shortest distance queries are one of the most fundamental graph operations and have a wide range of applications. Using such graph encryption schemes, a client can outsource large-scale privacy-sensitive graphs to an untrusted server without losing the ability to query it. Other applications include encrypted graph databases and controlled disclosure systems. We propose **GRECS** (stands for **GR**aph **E**n**C**ryption for approximate **S**hortest distance queries) which includes three schemes that are provably secure against any semi-honest server. Our first construction makes use of only symmetric-key operations, resulting in a computationally-efficient construction. Our second scheme, makes use of somewhat-homomorphic encryption and is less computationally-efficient but achieves optimal communication complexity (i.e., uses a minimal amount of bandwidth). Finally, our third scheme is both computationally-efficient and achieves optimal communication complexity at the cost of a small amount of additional leakage. We implemented and evaluated the efficiency of our constructions experimentally. The experiments demonstrate that our schemes are efficient and can be applied to graphs that scale up to $1.6$ million nodes and $11$ million edges.



## Graph Database Encryption Scheme

A graph encryption scheme for distance queries Graph = (Setup, DistQuery) consists of a polynomial-time algorithm and a polynomial-time two-party protocol that work as follows:

- $(K, \mathsf{EGR}) \leftarrow \mathsf{Setup}(1^k, G, \alpha, \varepsilon)$.

  $(d, \perp) \leftarrow \mathsf{distQuery}_{C,S}((K, q), \mathsf{EGR})$

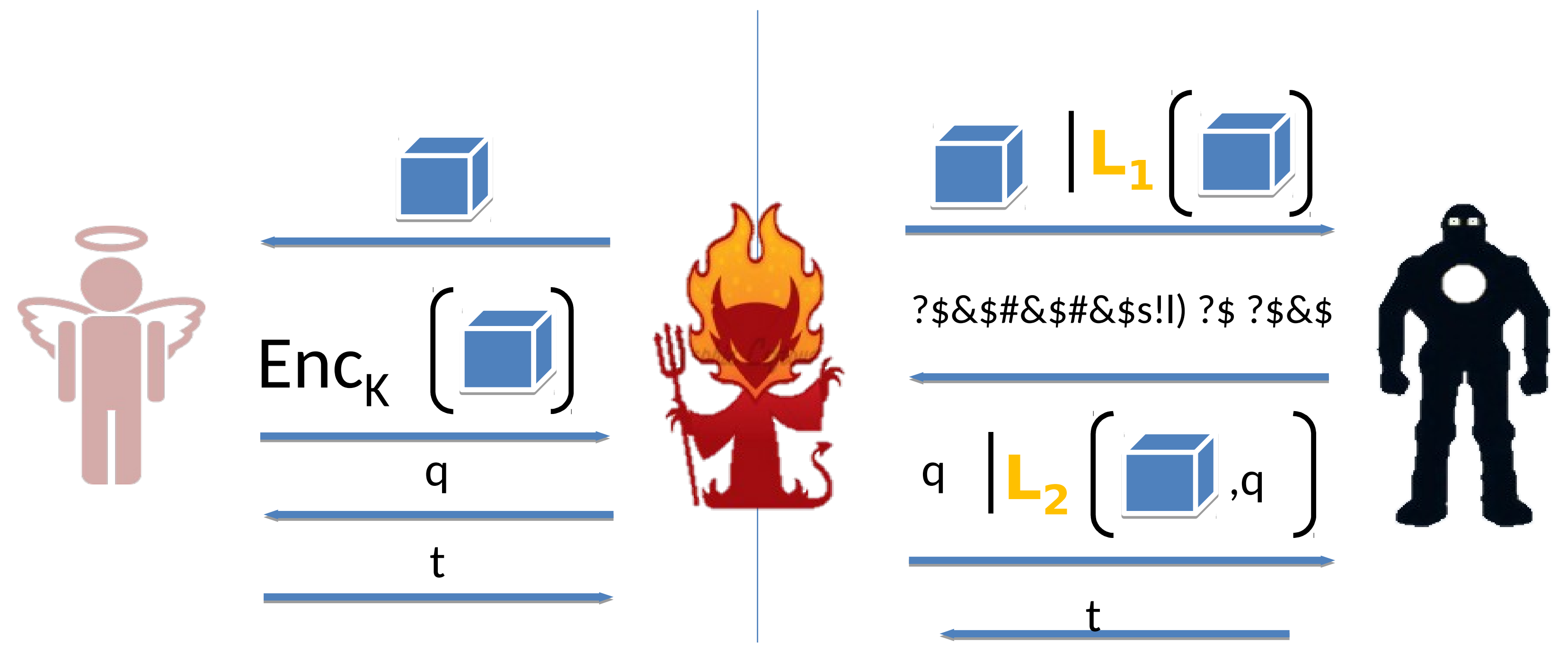
*We say that* Graph *is* $(\alpha, \varepsilon)$*-correct if for all* $k \in \mathbb{N}$, *for all* $G$, *for all* $\alpha \geq 1$, *for all* $\varepsilon < 1$, *and for all* $q = (u, v) \in V^2$,

$$\Pr[\, d \leq \alpha \cdot \mathsf{dist}(u, v)\,] \geq 1 - \varepsilon,$$

*where the probability is over the randomness in computing* $(K, \mathsf{EGR}) \leftarrow \mathsf{Setup}(1^k, G, \alpha, \varepsilon)$ *and then* $(d, \perp) \leftarrow$ $\mathsf{distQuery}((K, q), \mathsf{EGR})$.

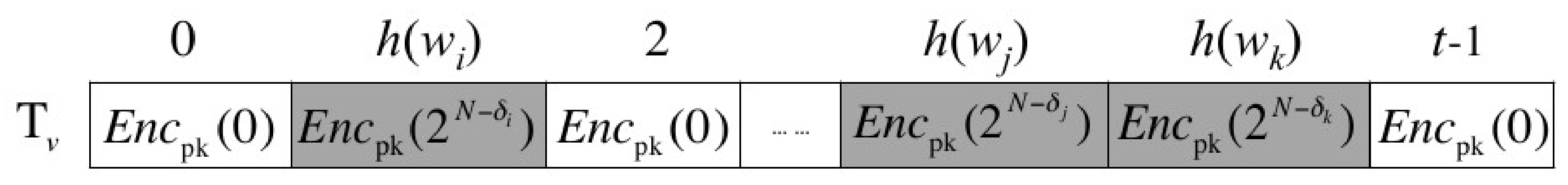
## Graph Database Security

**Security Definition** *(at a high level):*
No efficient adversary can learn any partial information about the graph or the queries, beyond what is explicitly allowed by the leakage functions. This holds even for queries that are adversarially-influenced and generated adaptively; that is, as a function of the encrypted graph and previous results.



## Efficient Graph Encryption Construction

We propose three constructions. Our first scheme only makes use of symmetric-key operations and, as such, is very computationally efficient. Our second scheme makes use of somewhat-homomorphic encryption (**BGN cryptosystem**) and achieves optimal communication complexity. Our third scheme is computationally-ecient, achieves optimal communication complexity and produces compact encrypted graphs at the cost of some leakage. We show that all our constructions are adaptively semantically-secure with reasonable leakage functions.

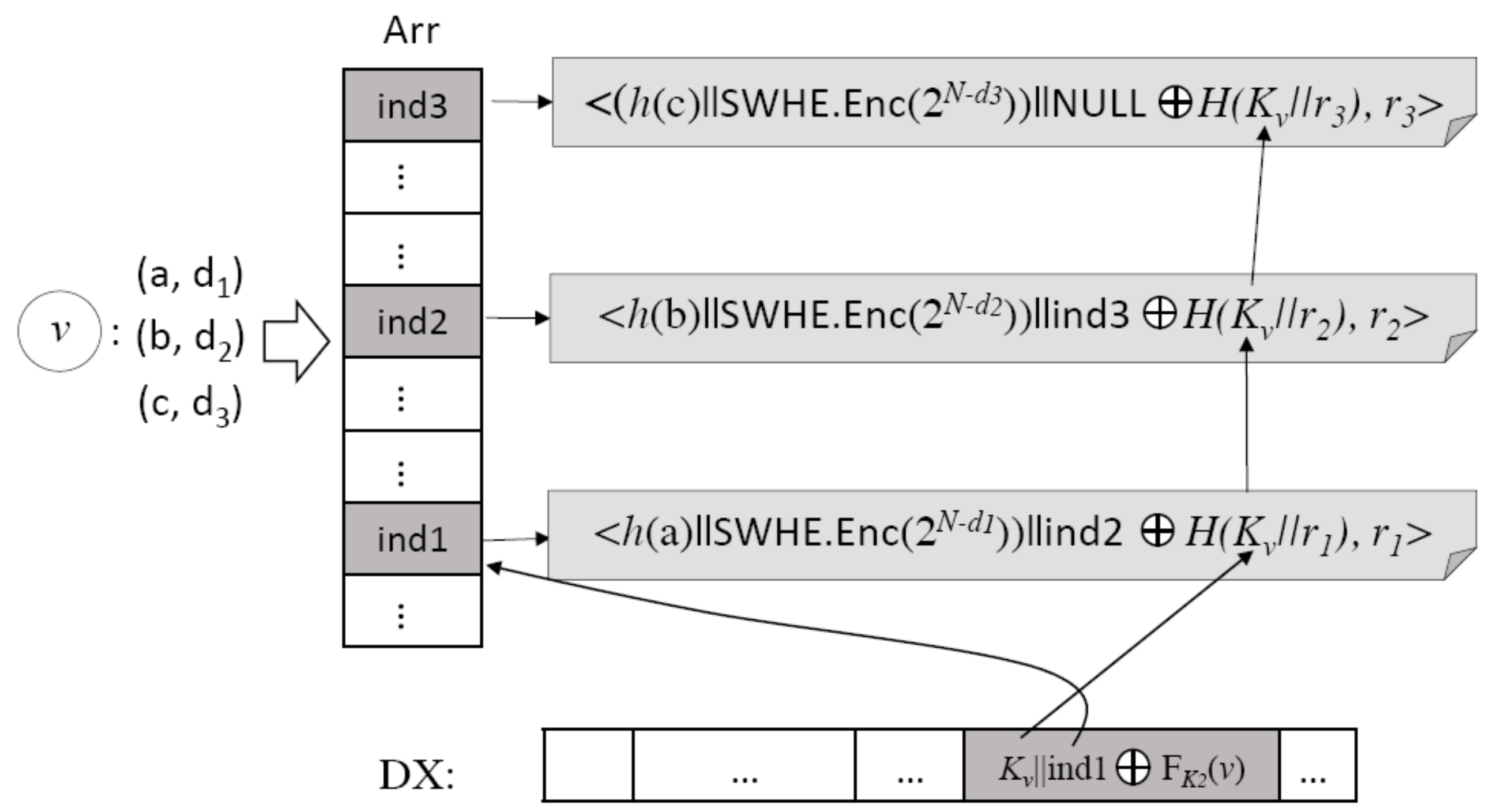


GraphEnc2: Communication-Efficient Construction

**Theorem** *Let* $G = (V, E)$, $\alpha \geq 1$ *and* $\varepsilon < 1$. *For all* $q = (u, v) \in V^2$ *with* $u \neq v$,

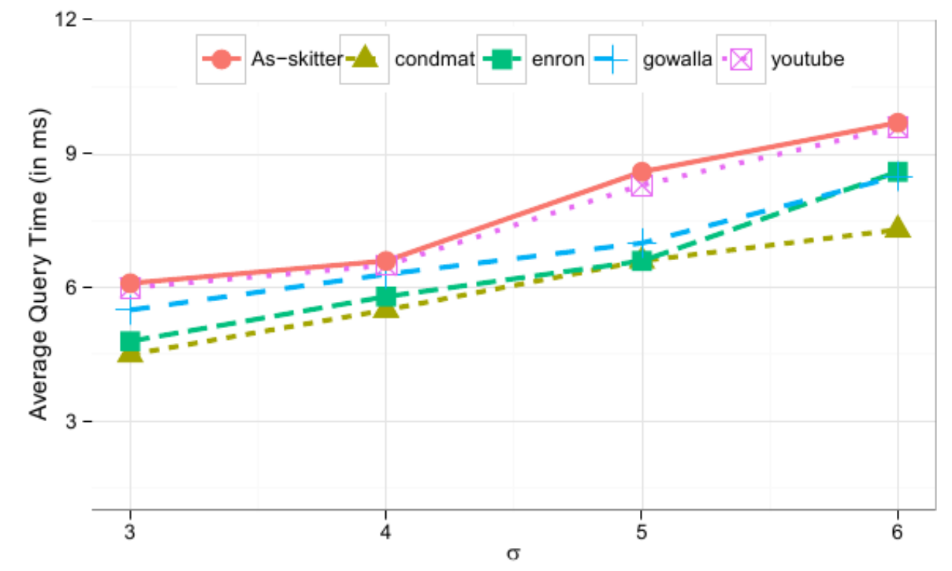
$$\Pr[\, d \leq \alpha \cdot \mathsf{dist}(u, v)\,] \geq 1 - \varepsilon,$$

*where* $(d, \perp) := \mathsf{GraphEnc_2.distQuery}((K, q), \mathsf{EGR})$ *and* $(K, \mathsf{EGR}) \leftarrow$ $\mathsf{GraphEnc_2.Setup}(1^k, G, \alpha, \varepsilon)$.

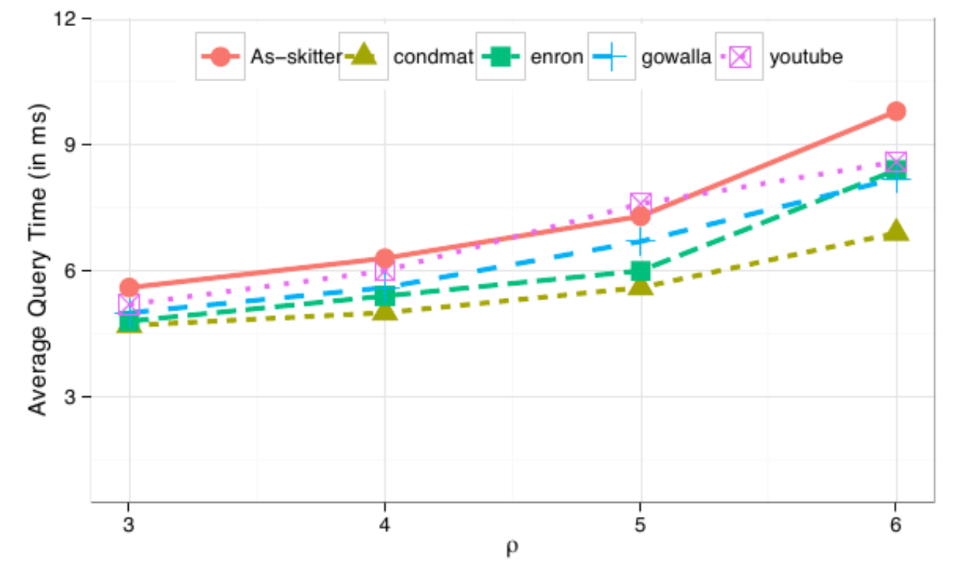

Arr
ind3
ind2
ind1
<(h(c)||SWHE.Enc(2^{N-d3}))||NULL ⊕H(K_v||r_3), r_3>
<h(b)||SWHE.Enc(2^{N-d2}))||ind3 ⊕H(K_v||r_2), r_2>
<h(a)||SWHE.Enc(2^{N-d1}))||ind2 ⊕ H(K_v||r_1), r_1>
v
(a, d$_1$)
(b, d$_2$)
(c, d$_3$)
DX:
...
...
K$_v$||ind1 ⊕ F$_{K2}$(v)
...

GraphEnc3:Space Efficient & Communication-Efficient Construction

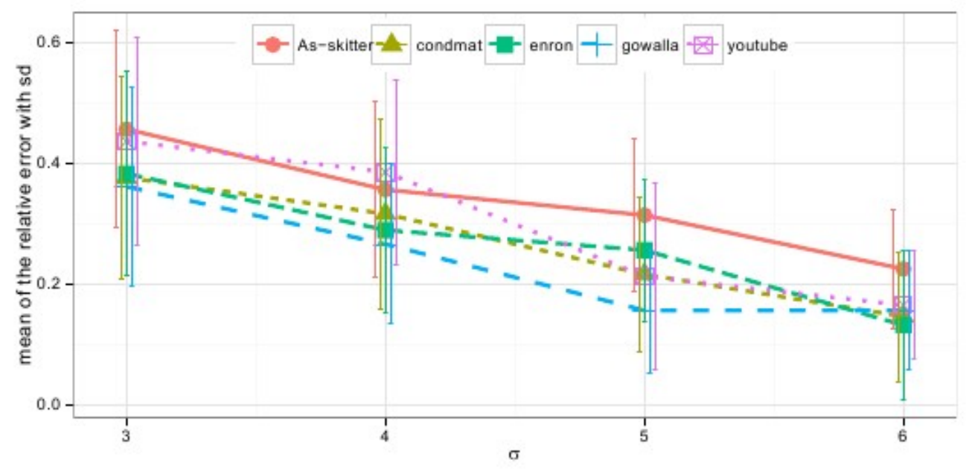
## Some Experimental Results



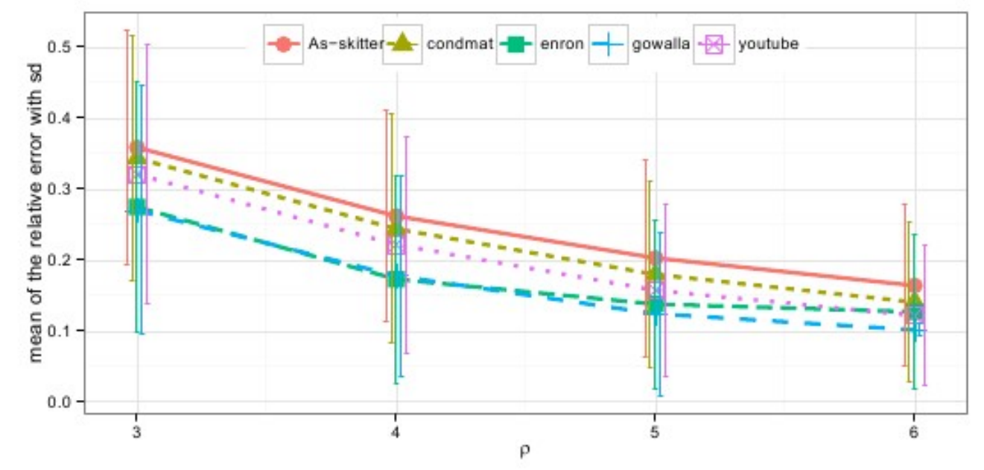
(a) Query Time (in ms) using $\mathbf{DO}_1$



(b) Query Time (in ms) $\mathbf{DO}_2$



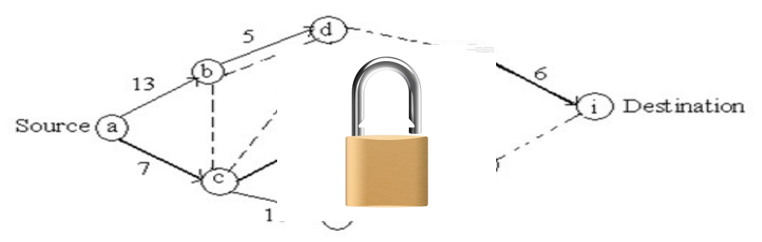
Mean of Estimated Error with Standard Deviation using $\mathbf{DO}_1$



Mean of Estimated Error with Standard Deviation using $\mathbf{DO}_2$

## However, how to compute shortest distance??

### Dijkstra, Bellman-Ford, Adj-Matrix? NO!



| | |
|---|---|
| $\mathsf{Sk}(v_i)$: | $\{(a,3),\ (b,3),\ (e,6),\ (g,3),\ (h,4)\}$ |
| $\mathsf{Sk}(v_j)$: | $\{(b,2),\ (d,1),\ (e,3),\ (h,3),\ (f,7)\}$ |

Figure 1: Two example sketches for nodes $v_i$ and $v_j$. The approximate shortest distance $d = 5$, since $b$ is in both sketches and the sum of its distances to $v_i$ and $v_j$ is the minimum sum.

## Sketch-based oracle!

**Sketched-based oracles** More formally, a sketch-based distance oracle $\mathsf{DO} = (\mathsf{Setup}, \mathsf{Query})$ is a pair of efficient algorithms that work as follows. $\mathsf{Setup}$ takes as input a graph $G$, an approximation factor $\alpha$ and an error bound $\varepsilon$ and outputs an oracle $\Omega_G = \{\mathsf{Sk}_v\}_{v \in V}$. $\mathsf{Query}$ takes as input an oracle $\Omega_G$ and a shortest distance query $q = (u, v)$. We say that $\mathsf{DO}$ is $(\alpha, \varepsilon)$-correct if for all graphs $G$ and all queries $q = (u, v)$,

$$\Pr[\, d \leq \alpha \cdot \mathsf{dist}(u, v)\,] \geq 1 - \varepsilon,$$

where $d := \mathsf{Query}(\Omega_G, u, v)$.