Supplemental Material: Optimal String Sanitization Against Strategic Attackers

Pengxin Bian, *Graduate Student Member, IEEE*, George Theodorakopoulos, Solon P. Pissis, and Grigorios Loukides, *Senior Member, IEEE*

I. Additional Related Work

There are two main directions in privacy-preserving data publishing; data sanitization and anonymization. The former has been reviewed in Section II of the main paper. The latter aims to prevent the inference of information about individuals represented in a disseminated dataset [1]. Below we discuss data anonymization approaches as a supplement to Section II of the main paper.

Instead of protecting the privacy of sensitive patterns modeling confidential information as data sanitization does, data anonymization aims to protect information about individuals. For example, anonymization aims to prevent the disclosure of individuals' identities or attributes these individuals are not willing to be associated with. Existing string anonymization approaches propose algorithms based on two privacy principles: k-anonymity [2] and differential privacy [3].

To enforce k-anonymity, several algorithms [4]–[6] apply *condensation*, a methodology which splits a dataset into multiple groups of size at least k and then releases aggregate statistics about each group. These algorithms aim to create groups of similar entities, so that the quality loss incurred by releasing the group statistics is small. For example, [5], [6] are applied to a collection of strings and create groups of at least k similar strings. They represent each group of strings using summary statistics containing first and second order information about the distribution of the letters in the strings, while [6] also discusses the possibility of releasing third order information.

The algorithms based on differential privacy could be classified into those that are applied to a collection of strings and those that are applied to a single string. [7] proposes an algorithm to extract differentially-private variable-length n-grams with counts larger than a threshold and lengths larger than another threshold from a collection of strings. [8] releases a set of differentially-private top-k frequent strings from a collection of strings, where k denotes the number of frequent strings required, while [9] releases a set of differentially-private frequent subsequences from a collection of strings under a maximum gap constraint. The approach of [10] proposes a sampling-based mechanism to enforce differential privacy on a single string and several algorithms, exact and heuristics, to achieve this while preserving data utility. The privacy goal of this approach is to prevent an attacker from inferring the presence in the disseminated string or absence from the disseminated string of any q-gram.

Since anonymization algorithms do not aim to hide sensitive patterns, they are not alternatives to our algorithms.

P. Bian and G. Loukides are with Department of Informatics, King's College London, 30 Aldwych, London WC2B 4BG, UK.

E-mail: {pengxin.bian, grigorios.loukides}@kcl.ac.uk

G. Theodorakopoulos is with Cardiff University, UK. Email: theodorakopoulosg@cardiff.ac.uk

S. P. Pissis is with CWI and the Vrije Universiteit, The Netherlands. Email: solon.pissis@cwi.nl

Manuscript received April 19, 2005; revised August 26, 2015.

II. Contexts Construction

Given a sensitive pattern $s \in S$, we find all its occurrences in the reference string $\mathcal{R} = \mathcal{R}[1] \cdot \ldots \cdot \mathcal{R}[n]$ using the pattern matching algorithm of [11]. Then, for each occurrence (starting position i) of s, we assign $c_l^\ell = \mathcal{R}[i-l] \cdot \ldots \cdot \mathcal{R}[i-1]$ and $c_l^r = \mathcal{R}[i+1] \cdot \ldots \cdot \mathcal{R}[i+l]$. If any of the strings c_l^ℓ or c_l^r overlaps with a sensitive pattern, we truncate it by removing the minimum number of letters from it so that it does not. This is to allow our algorithms to replace a pattern occurring right after c_l^ℓ and right before c_l^r without replacing letters of c_l^ℓ or c_l^r. An alternative would be to extend c_l^ℓ or c_l^r, so that it includes the sensitive pattern(s) that overlap with $\mathcal{R}[i-l] \cdot \ldots \cdot \mathcal{R}[i-1]$ or $\mathcal{R}[i+1] \cdot \ldots \cdot \mathcal{R}[i+l]$. We will henceforth assume the first way, as the second one did not lead to a practical benefit, based on our experiments. Furthermore, we truncate c_l^ℓ if $i-l<1$ or c_l^r if $i+l>|\mathcal{R}|$, to ensure that the context occurs in $\mathcal{R}$.

III. Best K Nonsensitive Patterns Selection

We use the pattern matching algorithm of [11] to find $\mathtt{occ}_{\mathcal{R}}(c_l^\ell)$ and $\mathtt{occ}_{\mathcal{R}}(c_l^r)$ (i.e., the positions of c_l^ℓ and of c_l^r in $\mathcal{R}$). Then, for each position $i \in \mathtt{occ}_{\mathcal{R}}(c_l^\ell)$, we find the closest position $j > i+|c_l^\ell|-1$ such that $j \in \mathtt{occ}_{\mathcal{R}}(c_l^r)$ (i.e., the closest starting position of c_l^r). Recall that U_{c_l} at this point contains all sensitive patterns that are associated with context c_l. If the string u' right after c_l^ℓ and right before c_l^r is not sensitive (i.e., $u' = \mathcal{R}[i+|c_l^\ell|] \cdot \ldots \cdot \mathcal{R}[j-1] \notin S$), then we add the pair $(\sum_{u \in U_{c_l}} d(u', u), u)$ into an initially empty priority queue, which is sorted in descending order of the first element of the pairs in it. As we consider the positions in $\mathtt{occ}_{\mathcal{R}}(c_l^\ell)$, the priority queue will contain K elements at some point[1]. Then, when we consider the current u', we

[1]If this never happens, we cannot have enough nonsensitive pattern replacements for c_l, so we report failure.

compute $\sum_{u \in U_{c_l}} d(u', u)$ incrementally (i.e., term by term) and if at some point the current sum exceeds the sum of the top element in the priority queue, we skip this u', as it cannot be one of the K best, and continue with the next one. Otherwise, the fully computed sum and this u form a pair which replaces the current top pair in the priority queue. Thus, the priority queue will have K elements, after all positions in $\mathtt{occ}_{\mathcal{R}}(c_l^{\ell})$ are considered. Last, we remove each element from the priority queue and add its string into the set U'_{c_l}.

IV. Linearization of the program for AQL-MILP

To prove that the linear program in Program 2 is equivalent to that in Program 1, we show below that, for any u', $z_{u'}$ indeed takes the minimum value of T_{MIN} and $M \cdot \sum_u P(u' \mid u)$.

$$\textbf{Minimize} \quad \sum_u \pi(u \mid c_l) \sum_{u'} P(u' \mid u) d_q(u', u) \tag{1}$$

subject to

$$\sum_u \pi(u \mid c_l) P(u' \mid u) d_p(\hat{u}, u) \geq \min\{T_{\text{MIN}}, M \cdot \sum_u P(u' \mid u)\}, \forall \hat{u}, u' \tag{2}$$

$$\sum_{u'} P(u' \mid u) = 1, \forall u \tag{3}$$

$$P(u' \mid u) \geq 0, \forall u, u' \tag{4}$$

Program 1: Average quality-loss-minimizing, privacy-constrained nonlinear program

$$\textbf{Minimize} \quad \sum_u \pi(u \mid c_l) \sum_{u'} P(u' \mid u) d_q(u', u) \tag{5}$$

subject to

$$z_{u'} \leq \sum_u \pi(u \mid c_l) P(u' \mid u) d_p(\hat{u}, u), \forall \hat{u}, u' \tag{6}$$

$$z_{u'} \leq T_{\text{MIN}}, \forall u' \tag{7}$$

$$z_{u'} \geq T_{\text{MIN}} - m \cdot y_{u',1}, \forall u' \tag{8}$$

$$z_{u'} \leq M \cdot \sum_u P(u' \mid u), \forall u' \tag{9}$$

$$z_{u'} \geq M \cdot \sum_u P(u' \mid u) - m \cdot y_{u',2}, \forall u' \tag{10}$$

$$y_{u',1} + y_{u',2} \leq 1, \forall u' \tag{11}$$

$$y_{u',1}, y_{u',2} \in \{0, 1\}, \forall u' \tag{12}$$

$$\sum_{u'} P(u' \mid u) = 1, \forall u \tag{13}$$

$$P(u' \mid u) \geq 0, \forall u, u' \tag{14}$$

Program 2: Average quality-loss-minimizing, privacy-constrained linear program

Case 1: $y_{u',1} = 0$**,** $y_{u',2} = 0$**.**
In Case 1, Eq. 8 becomes

$$z_{u'} \geq T_{\text{MIN}} \tag{15}$$

and Eq. 10 becomes

$$z_{u'} \geq M \cdot \sum_u P(u' \mid u). \tag{16}$$

From Eq. 7 and Eq. 15, we infer that

$$z_{u'} = T_{\text{MIN}} \tag{17}$$

From Eq. 9 and Eq. 16, we infer that

$$z_{u'} = M \cdot \sum_u P(u' \mid u) \tag{18}$$

From Eq. 17 and Eq. 18, we get

$$T_{\text{MIN}} = M \cdot \sum_u P(u' \mid u) \tag{19}$$

Therefore, in Case 1 $z(u')$ could be either T_{MIN} or $M \cdot \sum_u P(u' \mid u)$ for a specific u' as they are equal.

Case 2: $y_{u',1} = 0$**,** $y_{u',2} = 1$**.**
In Case 2, Eq. 8 becomes

$$z_{u'} \geq T_{\text{MIN}} \tag{20}$$

and Eq. 10 becomes

$$z_{u'} \geq M \cdot \sum_u P(u' \mid u) - m. \tag{21}$$

From Eq. 7 and Eq. 20, we infer that

$$z_{u'} = T_{\text{MIN}} \tag{22}$$

Eq. 9 and Eq. 21 will always hold, because $m - M \cdot \sum_u P(u' \mid u) > 0$ for every u' always holds.
Therefore, in Case 2 $z_{u'} = T_{\text{MIN}}$.

Case 3: $y_{u',1} = 1$**,** $y_{u',2} = 0$**.**
In Case 3, Eq. 8 becomes

$$z_{u'} \geq T_{\text{MIN}} - m \tag{23}$$

and Eq. 10 becomes

$$z_{u'} \geq M \cdot \sum_u P(u' \mid u). \tag{24}$$

Eq. 7 and Eq. 23 will always hold because $T_{\text{MIN}} - m < 0$ always holds.
From Eq. 9 and Eq. 24, we infer that

$$z_{u'} = M \cdot \sum_u P(u' \mid u) \tag{25}$$

Therefore, in Case 3 $z_{u'} = M \cdot \sum_u P(u' \mid u), \forall u'$.

These are the only cases, due to Eq. 11. Thus, the linearization of the minimum operator in AQL-MILP is correct; $z_{u'}$ will always be the minimum one among $T_{\text{MIN}}, M \cdot \sum_u P(u' \mid u)$ for every $u' \in U'_{c_l}$.

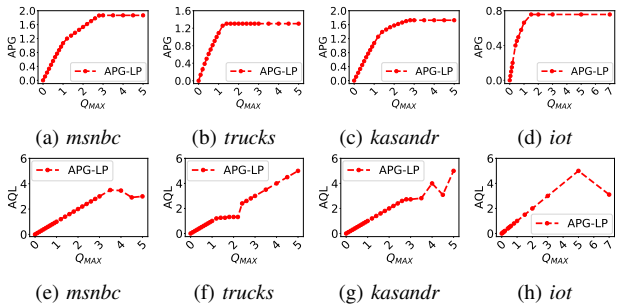
V. Additional Experimental Results

A. Impact of main parameters

Figs. 1, 2, 3, and 4 present the results of APG and AQL, illustrating how each parameter (Q_{MAX} for APG-LP, T_{MIN} for AQL-MILP, ϵ for DP-LP, and R for the Baseline) affects its respective algorithm on the *msnbc*, *trucks*, *kasandr*, and *iot* datasets. They correspond to the experiments in the Impact of Main Parameters part of Section IV-D in the main paper.

Figs. 5, 6, 7, and 8 show the APG and AQL of each method for varying K on the *msnbc*, *trucks*, *kasandr*, and *iot* datasets. They correspond to the experiments in the Impact of Main Parameters part of Section IV-D in the main paper.

Figs. 9, 10, 11, 12, and 13 show the results of APG and AQL of each method for varying l on the *iot*, *ecoli*, *msnbc*, *trucks*, and *kasandr* datasets. The results are too data-dependent to draw general conclusions (e.g., they depend on how the values of the d_q matrix and $\pi(u \mid c_l)$ values change). Take *iot* dataset as an example. Fig. 9 shows that increasing l results in a lower AQL for all algorithms, as the values in their d_q matrices decrease. As l increases, the APG-LP algorithm has a higher privacy (APG increases); see Fig. 9a. This is because the differences between the probabilities $\pi(u \mid c_l)$ for the different strings $u \in U_{c_l}$ become smaller, while the values in the d_p matrix do not change (observe Eqs. 8 and 9 in the main paper). On the other hand, AQL-MILP has the same APG (equal to $T_{\text{MIN}} \cdot |U_{c_l}|$); see Fig. 9c. For DP-LP and Baseline, APG decreases (see Figs. 9e and 9g) because the $\pi(u \mid c_l)$ values change as explained above and the values in d_p stay the same (see Eqs. 5 and 6). On the other hand, for the *ecoli* dataset, increasing l results in a generally higher AQL for all algorithms (see Figs. 10b, 10d, 10f, and 10h), as the values in their d_q matrices increase, while APG stays the same (see Figs. 10a, 10c, 10e, and 10h) because in this case $\pi(u \mid c_l)$ does not vary with l, nor does the d_p matrix.



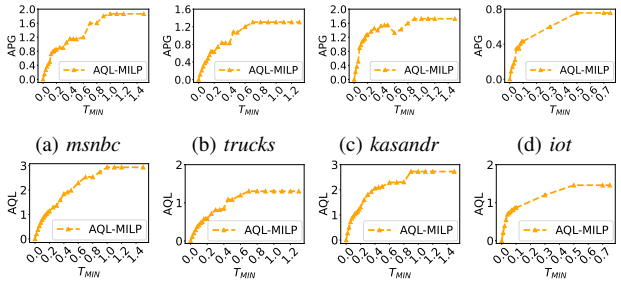
(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

Fig. 1: (a)(b)(c)(d) APG and (e)(f)(g)(h) AQL for APG-LP vs. Q_{MAX} in the *msnbc*, *trucks*, *kasandr*, and *iot* datasets.

B. Hybrid Algorithms

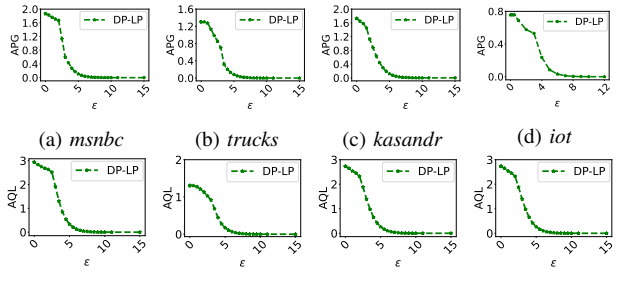
Recall that the first hybrid algorithm APG-MILP maximizes APG while ensuring that AQL does not exceed Q_{MAX} and that each possible replacement u' gains at least T_{MIN} privacy. Its mathematical programming formulation is shown in Program 3.



(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

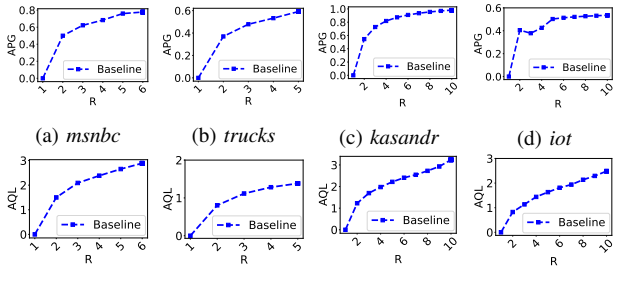
Fig. 2: (a)(b)(c)(d) APG and (e)(f)(g)(h) AQL for AQL-MILP vs. T_{MIN} in the *msnbc*, *trucks*, *kasandr*, and *iot* datasets.



(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

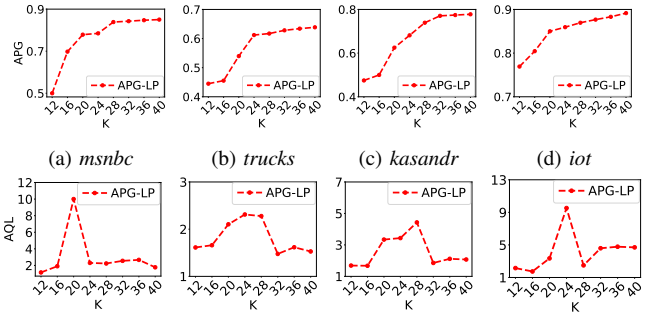
Fig. 3: (a)(b)(c)(d) APG and (e)(f)(g)(h) AQL for AQL-MILP vs. ϵ in the *msnbc*, *trucks*, *kasandr*, and *iot* datasets.



(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

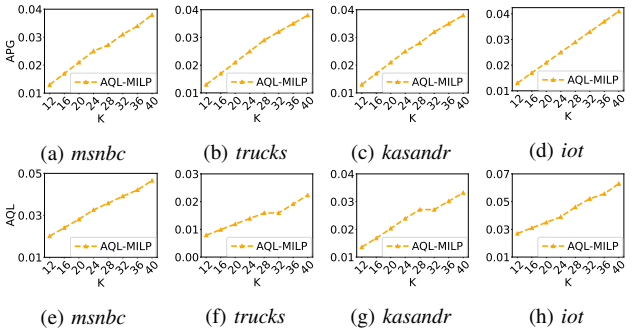
Fig. 4: (a)(b)(c)(d) APG and (e)(f)(g)(h) AQL for AQL-MILP vs. R in the *msnbc*, *trucks*, *kasandr*, and *iot* datasets.



(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

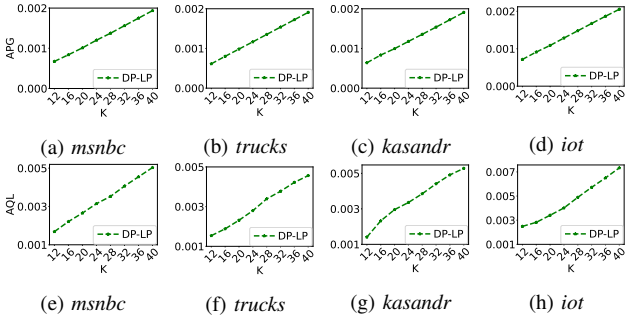
Fig. 5: APG and AQL for APG-LP vs. K in the *msnbc*, *trucks*, *kasandr*, and *iot* dataset.



(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

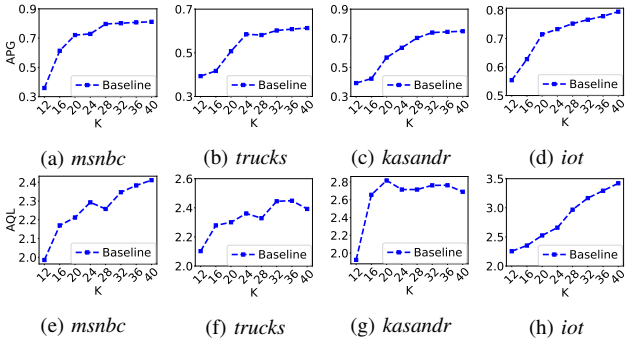
Fig. 6: APG and AQL for AQL-MILP vs. K in the *msnbc*, *trucks*, *kasandr*, and *iot* dataset.



(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

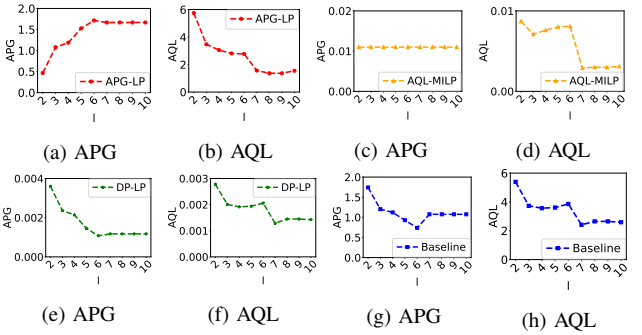
Fig. 7: APG and AQL for DP-LP vs. K in the *msnbc*, *trucks*, *kasandr*, and *iot* dataset.



(a) *msnbc* (b) *trucks* (c) *kasandr* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *kasandr* (h) *iot*

Fig. 8: APG and AQL for Baseline vs. K in the *msnbc*, *trucks*, *kasandr*, and *iot* dataset.

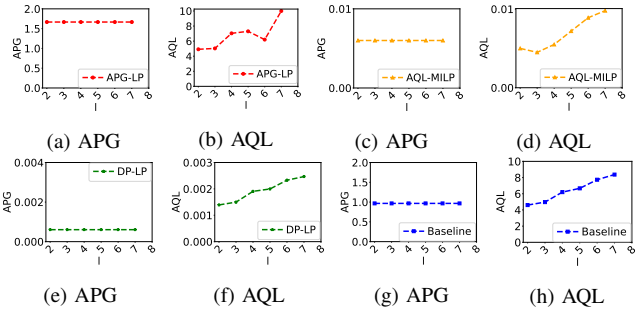


(a) APG (b) AQL (c) APG (d) AQL

(e) APG (f) AQL (g) APG (h) AQL

Fig. 9: (a)(c)(e)(g) APG and (b)(d)(f)(h) AQL vs. l for APG-LP, AQL-MILP, DP-LP, and Baseline. All results are for the *iot* dataset.

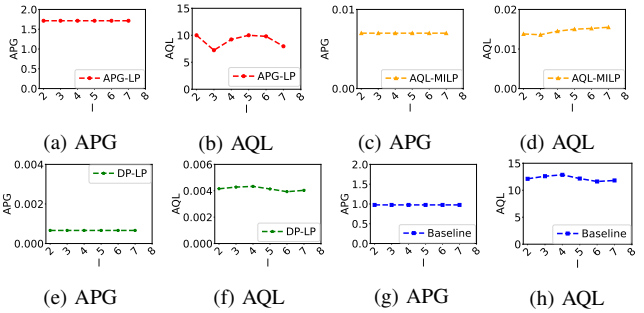
Fig. 14 and Fig. 15 show the results of WAPG, WMPG



(a) APG (b) AQL (c) APG (d) AQL

(e) APG (f) AQL (g) APG (h) AQL

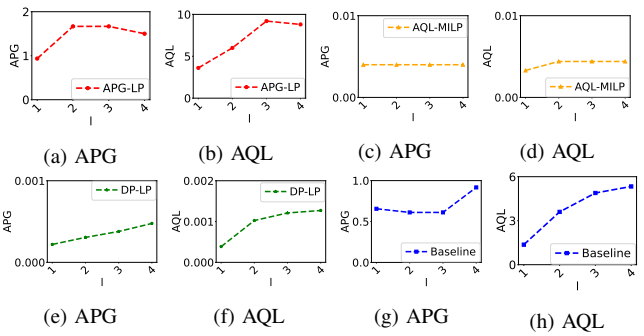
Fig. 10: (a)(c)(e)(g) APG and (b)(d)(f)(h) AQL vs. l for APG-LP, AQL-MILP, DP-LP, and Baseline. All results are for the *ecoli* dataset.



(a) APG (b) AQL (c) APG (d) AQL

(e) APG (f) AQL (g) APG (h) AQL

Fig. 11: (a)(c)(e)(g) APG and (b)(d)(f)(h) AQL vs. l for APG-LP, AQL-MILP, DP-LP, and Baseline. All results are for the *msnbc* dataset.



(a) APG (b) AQL (c) APG (d) AQL

(e) APG (f) AQL (g) APG (h) AQL

Fig. 12: (a)(c)(e)(g) APG and (b)(d)(f)(h) AQL vs. l for APG-LP, AQL-MILP, DP-LP, and Baseline. All results are for the *trucks* dataset.

and WAQL when varying Q_{MAX} and T_{MIN} respectively in *msnbc*, *trucks*, *ecoli*, and *iot* datasets, which correspond to the experiments in the Hybrid Algorithms part of Section IV-D in the main paper.

Recall that the second hybrid algorithm, DP-MILP, is derived by incorporating the differentially private constraint from DP-LP into AQL-MILP. The mathematical programming formulation for DP-MILP is shown in Program 4.

Similar to APG-MILP, we evaluate DP-MILP by varying T_{MIN} and ϵ separately. Fig. 16 and Fig. 17 show the results of WAPG and WAQL when varying T_{MIN} and ϵ respectively on the *msnbc*, *trucks*, *ecoli*, and *iot* datasets. These results supplement the experiments discussed in the Hybrid Algo-

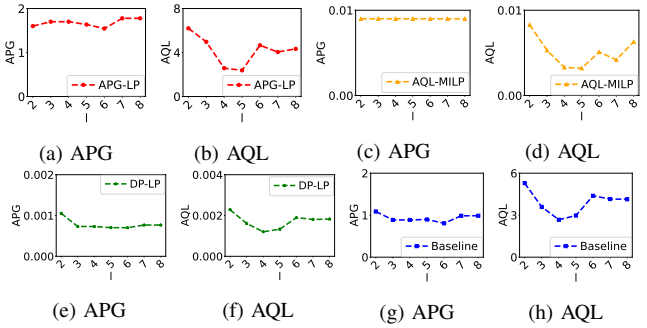


Fig. 13: (a)(c)(e)(g) APG and (b)(d)(f)(h) AQL vs. l for APG-LP, AQL-MILP, DP-LP, and Baseline. All results are for the *kasandr* dataset.

$$\textbf{Maximize} \quad \sum_{u'} z_{u'} \tag{26}$$

subject to

$$\sum_{u} \pi(u \mid c_l) \sum_{u'} P(u' \mid u) d_q(u', u) \leq Q_{\text{MAX}} \tag{27}$$

$$z_{u'} \leq \sum_{u} \pi(u \mid c_l) P(u' \mid u) d_p(\hat{u}, u), \forall \hat{u}, u' \tag{28}$$

$$z_{u'} \leq T_{\text{MIN}}, \forall u' \tag{29}$$

$$z_{u'} \geq T_{\text{MIN}} - m \cdot y_{u',1}, \forall u' \tag{30}$$

$$z_{u'} \leq M \cdot \sum_{u} P(u' \mid u), \forall u' \tag{31}$$

$$z_{u'} \geq M \cdot \sum_{u} P(u' \mid u) - m \cdot y_{u',2}, \forall u' \tag{32}$$

$$y_{u',1} + y_{u',2} \leq 1, \forall u' \tag{33}$$

$$y_{u',1}, y_{u',2} \in \{0, 1\}, \forall u' \tag{34}$$

$$\sum_{u'} P(u' \mid u) = 1, \forall u \tag{35}$$

$$P(u' \mid u) \geq 0, \forall u, u' \tag{36}$$

Program 3: APG-MILP: Combination of APG-LP and AQL-MILP to maximize APG.

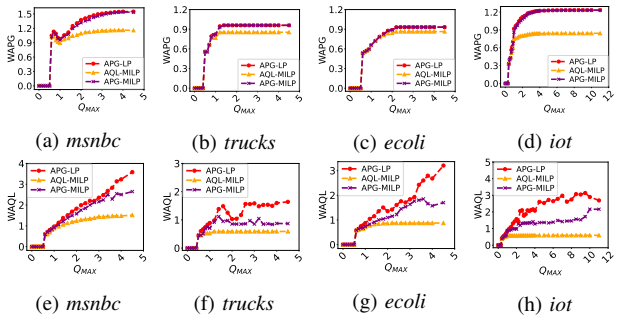


Fig. 14: WAPG and WAQL for APG-MILP vs. Q_{MAX} in the *msnbc*, *trucks*, *ecoli*, and *iot* datasets.

rithms part of Section IV-D of the main paper.

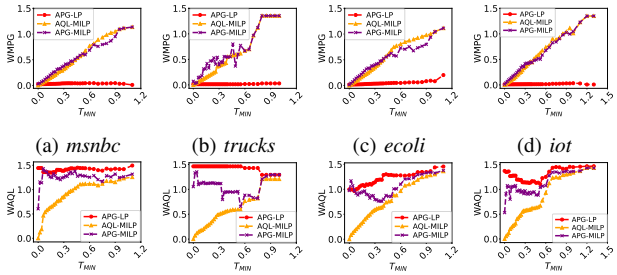


Fig. 15: WMPG and WAQL for APG-MILP vs. T_{MIN} in the *msnbc*, *trucks*, *ecoli*, and *iot* datasets.

$$\textbf{Minimize} \quad \sum_{u} \pi(u \mid c_l) \sum_{u'} P(u' \mid u) d_q(u', u) \tag{37}$$

subject to

$$z_{u'} \leq \sum_{u} \pi(u \mid c_l) P(u' \mid u) d_p(\hat{u}, u), \forall \hat{u}, u' \tag{38}$$

$$z_{u'} \leq T_{\text{MIN}}, \forall u' \tag{39}$$

$$z_{u'} \geq T_{\text{MIN}} - m \cdot y_{u',1}, \forall u' \tag{40}$$

$$z_{u'} \leq M \cdot \sum_{u} P(u' \mid u), \forall u' \tag{41}$$

$$z_{u'} \geq M \cdot \sum_{u} P(u' \mid u) - m \cdot y_{u',2}, \forall u' \tag{42}$$

$$y_{u',1} + y_{u',2} \leq 1, \forall u' \tag{43}$$

$$P(u' \mid u_i) \leq P(u' \mid u_j) \cdot \mathrm{e}^{\epsilon}, \forall u' \in U'_{c_l}, u_i, u_j \in U_{c_l} \tag{44}$$

$$y_{u',1}, y_{u',2} \in \{0, 1\}, \forall u' \tag{45}$$

$$\sum_{u'} P(u' \mid u) = 1, \forall u \tag{46}$$

$$P(u' \mid u) \geq 0, \forall u, u' \tag{47}$$

Program 4: DP-MILP: Combination of AQL-MILP and DP-LP to minimize the AQL

REFERENCES

[1] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 14:1–14:53, 2010.

[2] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *Proc. ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst. (PODS)*, vol. 98, no. 188, 1998, pp. 10–1145.

[3] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf. (TCC)*, vol. 3876, 2006, pp. 265–284.

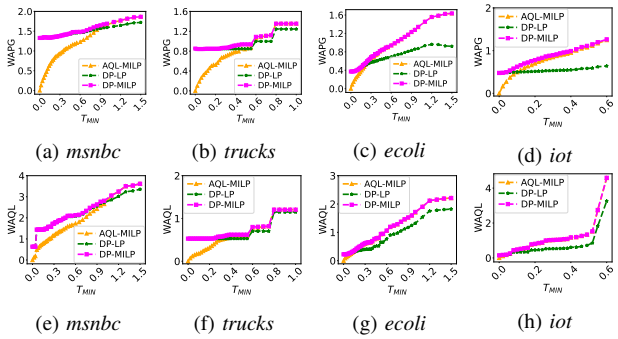
[4] C. C. Aggarwal and P. S. Yu, "A condensation approach to privacy preserving data mining," in *Proc. Int. Conf. Extending Database Technol. (EDBT)*, vol. 2992, 2004, pp. 183–199.

[5] ——, "On anonymization of string data," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2007, pp. 419–424.

[6] ——, "A framework for condensation-based anonymization of string data," *Data Min. Knowl. Discov.*, vol. 16, no. 3, pp. 251–275, 2008.

[7] R. Chen, G. Ács, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, 2012, pp. 638–649.

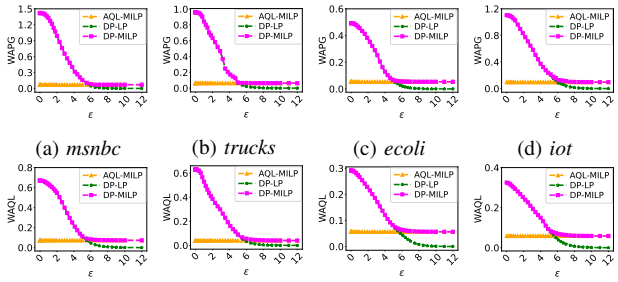
[8] L. Bonomi and L. Xiong, "A two-phase algorithm for mining sequential patterns with differential privacy," in *Proc. ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2013, pp. 269–278.



(a) *msnbc* (b) *trucks* (c) *ecoli* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *ecoli* (h) *iot*

Fig. 16: WAPG and WAQL for DP-MILP vs. T_{MIN} in the *msnbc*, *trucks*, *ecoli*, and *iot* datasets.



(a) *msnbc* (b) *trucks* (c) *ecoli* (d) *iot*

(e) *msnbc* (f) *trucks* (g) *ecoli* (h) *iot*

Fig. 17: WAPG and WAQL for DP-MILP vs. ϵ in the *msnbc*, *trucks*, *ecoli*, and *iot* datasets.

[9] S. Xu, X. Cheng, S. Su, K. Xiao, and L. Xiong, "Differentially private frequent sequence mining," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 11, pp. 2910–2926, 2016.

[10] H. Chen, C. Dong, L. Fan, G. Loukides, S. P. Pissis, and L. Stougie, "Differentially private string sanitization for frequency-based mining tasks," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2021, pp. 41–50.

[11] M. Crochemore, C. Hancart, and T. Lecroq, *Algorithms on strings*. Cambridge University Press, 2007.