Background

The Web is transitioning away from centralised services to a re-emergent decentralised platform. This generates demand for technologies that hide complexities of federated architectures (Verborgh, 2021) so developers can create rich Web 3.0 (Berners-Lee et al., 2001; Berners-Lee, 2001) applications.

Concurrently, privacy-preserving computation techniques are maturing. With greater processing power, secure multi-party computation (SMPC) (Cramer et al., 2015) has evolved from theoretical protocols (Yao, 1986) to applied algorithms and frameworks (Archer et al., 2018; Agahari et al., 2022). Further, techniques such as Fully Homomorphic Encryption (FHE) are approaching commercial viability for the cloud (Creeger, 2022). The potential of these techniques to protect data in decentralised applications is largely unrealised as scarce specialist knowledge is required to implement each use-case (Lindell, 2020).

Objectives

Research generalisable techniques for executing queries over *arbitrary* decentralised data by asking the question:

How can privacy-preserving computation techniques enhance decentralised query engines?

Specifically, how can engines account for privacy policies, notions of trust and the computational capacity of peers whilst shielding users from:

RO1 privacy-preserving algorithms: by automating algorithm selection during query planning, as we cannot expect Web developers to select or implement them - just as developers need not learn HTTPS encryption schemes;

RO2 privacy policies on datasets (Debackere et al., 2022): as applications should be decoupled from dataset constraints - dealing only with processed and abstracted facts;

RO3 data views and APIs (Dedecker et al., 2022): instead, querying an abstracted view - just as developers use URLs without managing DNS lookups to dereference them; and

RO4 the nature of the source datasets (Slabbinck et al., 2022): for instance, engines should determine if I can legally drink when my profile contains my exact date of birth *or* when it only contains my age as an integer.

RO1 is the core objective. **RO2**-**RO4** ensure the engine is robust across use-cases.

Outcomes

To investigate this, one could develop a novel *multi-agent query-planning* (MAQP) framework in which agents describe their privacy policies and query expressivity in a formalised logic. We propose the *arbitrary* data [**RO4**] and queries input to the system be encoded using RDF (Consortium et al., 2014), a mature self-describing data-model for information-exchange. An initial architecture could have the extended SPARQL API designed for centralised CQE (Cuenca Grau et al., 2013) which:

- *introspects* the *privacy policies* [**RO2**] of each relevant data-store and *query expressivity* [**RO3**] of each computing agent in the network;
- *plans* the query operations required by each agent, optimising to *maximise* the number of (sound) results produced and *minimise* the sensitive data shared between peers [**RO1**];
- contains *query-agents* capable of [**RO1**]:
    1. Secure Distributed Inner Joins (SDIJ) (Mohassel et al., 2020);
    2. SMPC over literals (Yao, 1986);
    3. FHE on cloud infrastructure (Gentry, 2009); and
    4. Dialogical Reasoning[1] (DR).
- *accounts for metadata* describing the reliability of data and query agents to produce confidence intervals for the correctness of results [**RO4**].

Plan

The research plan described below, first requires the development of a Universal Service Description[2] (USD) [**RO3**], Lowest Common Denominator[3] (LCD) [**RO3**] and Query Planning[4] (Dresselhaus et al., 2021; Riggan et al., 2019) (QP) [**RO1**] vocabularies. I plan to complete these collaboratively with SolidLab[5] and Inrupt[6] prior to commencing a DPhil.

1. Define use-cases and requirements in collaboration with academia and industry, and demonstrate these in a zero-privacy context using existing query and reasoning engines (Nenov et al., 2015; Verborgh and De Roo, 2015; Taelman et al., 2018; Berners-Lee et al., 2008)[7,8] (*2 months*).

[1] https://github.com/SolidLabResearch/Challenges/issues/22
[2] https://docs.google.com/document/d/1NL5SesXPzhAk0bSIEXjuUaySUY9D0kjWlnwo1V0fj24/
[3] https://docs.google.com/document/d/1iuv0y14oXeMdx2lt0NRq5knEZ9LLokDTfIU_N5k2qJw/
[4] https://docs.google.com/document/d/1JKcenbf0kvl60XjIb8XSuGKttS1QIhjVDuzpc0Ba5k0/
[5] https://solidlab.be/
[6] https://www.inrupt.com/
[7] https://github.com/comunica/comunica-feature-reasoning
[8] https://github.com/rdfjs/N3.js/pull/296

2. Develop rudimentary MAQPs, testing combinations of existing policy, query and reasoning profiles against query-agents implementing SDIJ, SMPC, FHE, DR and other privacy-preserving techniques (*6-12 months*).

3. Evaluate the MAPQs. Quantitatively, I will measure result quality (number of sound SELECT results and percentage of determinate ASK results) (Cuenca Grau et al., 2013), performance on varied network architectures[9] and computational cost. Qualitatively, I will assess the architectural complexity of implementing each privacy-preserving technique, including the level of "hard-coding" required to handle distinct data types and values. Specifically, I shall test the hypothesis that generic SMPC engines (Halevi et al., 2016) can be implemented by dereferencing machine-readable MPC algorithms (De Meester et al., 2016) that are indexed by the type signature of the function they evaluate (*1 month*).

4. Formalise a single logic for expressing policies, queries and reasoning profiles in MAQPs. As there is no consensus on a logic for the Web (Hayes, 2009), to withstand shifts in popular standards, this logic must subsume 'sensible' paradigms (as determined by the previous experiments). I expect these to include description logics (Baader et al., 2003) (DL), SWRL (Horrocks et al., 2004), RDF surfaces (De Roo and Hochstenbach, 2022) and RIF (Kifer, 2008). In formalising a general logic, I shall review existing logics including DL, modal logics (Chagrov, 1997) (which subsume DL and support meta-statements), first-order logic (Smullyan, 1995) (a modal logic with direct correspondence to RDF surfaces) and type theory (Martin-Löf and Sambin, 1984), where proofs as first-class citizens aid provenance (*3 months*).

5. Iterate on the USD, LSD and QP vocabularies to correspond with the above logic (*1 month*), generalise the rudimentary MAQPs to a single MAQP that uses this formalised logic (*4 months*) and evaluate it using the aforementioned measures (*0.5 months*).

6. Investigate provenance (Keskisärkkä et al., 2019) and probabilities (Keskisärkkä et al., 2020) in MAQPs by exchanging formal proofs with query results (Berners-Lee et al., 2008), in addition to associating confidence intervals to query results based on trust in other network members and source data (*6-12 months*). This should be evaluated using an extension of the aforementioned metrics (*0.5 months*).

7. Analyse the security implications of my work by:

    a) formally proving that data revealed during query execution cannot be used to reverse-engineer protected data (Grau et al., 2014; Sweeney, 2000);

    b) analysing how to limit query complexity and prevent denial of service attacks (Erling and Mikhailov; Kumar and Kumar, 2014); and

    c) investigating link-traversal exploits (Taelman and Verborgh, 2022) (*6 months*).

[9] https://github.com/SolidBench/SolidBench.js

Bibliography

Agahari, W.; Ofe, H.; and de Reuver, M., 2022. It is not (only) about privacy: How multi-party computation redefines control, trust, and risk in data sharing. *Electronic Markets*, 32, 3 (2022), 1577–1602. [Cited on page i.]

Archer, D. W.; Bogdanov, D.; Lindell, Y.; Kamm, L.; Nielsen, K.; Pagter, J. I.; Smart, N. P.; and Wright, R. N., 2018. From keys to databases—real-world applications of secure multi-party computation. *The Computer Journal*, 61, 12 (2018), 1749–1771. [Cited on page i.]

Baader, F.; Calvanese, D.; McGuinness, D.; Patel-Schneider, P.; Nardi, D.; et al., 2003. *The description logic handbook: Theory, implementation and applications*. Cambridge university press. [Cited on page iii.]

Berners-Lee, T., 2001. Web 3.0. (2001). [Cited on page i.]

Berners-Lee, T.; Connolly, D.; Kagal, L.; Scharf, Y.; and Hendler, J., 2008. N3logic: A logical framework for the world wide web. *Theory and Practice of Logic Programming*, 8, 3 (2008), 249–269. [Cited on pages ii and iii.]

Berners-Lee, T.; Hendler, J.; and Lassila, O., 2001. The semantic web. *Scientific american*, 284, 5 (2001), 34–43. [Cited on page i.]

Chagrov, A., 1997. Modal logic. (1997). [Cited on page iii.]

Consortium, W. W. W. et al., 2014. Rdf 1.1 concepts and abstract syntax. (2014). [Cited on page ii.]

Cramer, R.; Damg, I. B.; et al., 2015. *Secure multiparty computation*. Cambridge University Press. [Cited on page i.]

Creeger, M., 2022. The rise of fully homomorphic encryption: Often called the holy grail of cryptography, commercial fhe is near. *Queue*, 20, 4 (2022), 39–60. [Cited on page i.]

Cuenca Grau, B.; Kharlamov, E.; Kostylev, E. V.; and Zheleznyakov, D., 2013. Controlled query evaluation over owl 2 rl ontologies. In *International Semantic Web Conference*, 49–65. Springer. [Cited on pages ii and iii.]

De Meester, B.; Dimou, A.; Verborgh, R.; and Mannens, E., 2016. An ontology to semantically declare and describe functions. In *European Semantic Web Conference*, 46–49. Springer. [Cited on page iii.]

De Roo, J. and Hochstenbach, P. H., 2022. Rdf surfaces primer. (2022). [Cited on page iii.]

Debackere, L.; Colpaert, P.; Taelman, R.; and Verborgh, R., 2022. A policy-oriented architecture for enforcing consent in Solid. In *Proceedings of the 2nd International Workshop on Consent Management in Online Services, Networks and Things*, 516–524. Association for Computing Machinery. doi:10.1145/3487553.3524630. https://dl.acm.org/doi/pdf/10.1145/3487553.3524630. [Cited on page i.]

Dedecker, R.; Slabbinck, W.; Wright, J.; Hochstenbach, P.; Colpaert, P.; and Verborgh, R., 2022. What's in a pod? – a knowledge graph interpretation for the Solid ecosystem. In *Proceedings of the 6th Workshop on Storing, Querying and Benchmarking Knowledge Graphs*, vol. 3279 of *CEUR Workshop Proceedings*, 81–96. https://solidlabresearch.github.io/WhatsInAPod/. [Cited on page i.]

Dresselhaus, J.; Filippov, I.; Gengenbach, J.; Heling, L.; and Käfer, T., 2021. Slurp: An interactive sparql query planner. In *European Semantic Web Conference*, 15–20. Springer. [Cited on page ii.]

Erling, O. and Mikhailov, I. Faceted views over large-scale linked data. [Cited on page iii.]

Gentry, C., 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 169–178. [Cited on page ii.]

Grau, B. C.; Kharlamov, E.; Kostylev, E. V.; and Zheleznyakov, D., 2014. Controlled query evaluation over lightweight ontologies. In *Description Logics*, 141–152. Citeseer. [Cited on page iii.]

Halevi, S.; Ishai, Y.; Jain, A.; Kushilevitz, E.; and Rabin, T., 2016. Secure multiparty computation with general interaction patterns. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, 157–168. [Cited on page iii.]

Hayes, P. J., 2009. Blogic or now what's in a link. In *Keynote presented at the 8th International Semantic Web Conference, October, Washington, DC*. http://videolectures.net/iswc09_hayes_blogic. [Cited on page iii.]

Horrocks, I.; Patel-Schneider, P. F.; Boley, H.; Tabet, S.; Grosof, B.; Dean, M.; et al., 2004. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21, 79 (2004), 1–31. [Cited on page iii.]

Keskisärkkä, R.; Blomqvist, E.; Lind, L.; and Hartig, O., 2019. Rsp-ql*: Enabling statement-level annotations in rdf streams. In *Semantic Systems. The Power of AI and Knowledge Graphs*, 140–155. Springer International Publishing, Cham. [Cited on page iii.]

Keskisärkkä, R.; Blomqvist, E.; Lind, L.; and Hartig, O., 2020. Capturing and querying uncertainty in rdf stream processing. In *International Conference on Knowledge Engineering and Knowledge Management*, 37–53. Springer. [Cited on page iii.]

Kifer, M., 2008. Rule interchange format: The framework. In *International Conference on Web Reasoning and Rule Systems*, 1–11. Springer. [Cited on page iii.]

Kumar, S. and Kumar, S., 2014. Semantic web attacks and countermeasures. In *2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014)*, 1–5. IEEE. [Cited on page iii.]

Lindell, Y., 2020. Secure multiparty computation. *Commun. ACM*, 64, 1 (dec 2020), 86–96. doi:10.1145/3387108. https://doi.org/10.1145/3387108. [Cited on page i.]

Martin-Löf, P. and Sambin, G., 1984. *Intuitionistic type theory*, vol. 9. Bibliopolis Naples. [Cited on page iii.]

Mohassel, P.; Rindal, P.; and Rosulek, M., 2020. Fast database joins and psi for secret shared data. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 1271–1287. [Cited on page ii.]

Nenov, Y.; Piro, R.; Motik, B.; Horrocks, I.; Wu, Z.; and Banerjee, J., 2015. Rdfox: A highly-scalable rdf store. In *International Semantic Web Conference*, 3–20. Springer. [Cited on page ii.]

Riggan, T. et al., 2019. Using sparql explain to understand query execution in amazon neptune. *AWS Database Blog, https://aws. amazon. com/blogs/database/using-sparql-explain-to-understand-query-execution-in-amazon-neptune*, (2019). [Cited on page ii.]

Slabbinck, W.; Dedecker, R.; Vasireddy, S.; Verborgh, R.; and Colpaert, P., 2022. Event sourcing in solid. In *Proceedings of the 8th Workshop on Managing the Evolution and Preservation of the Data Web*. [Cited on page i.]

Smullyan, R. M., 1995. *First-order logic*. Courier Corporation. [Cited on page iii.]

Sweeney, L., 2000. Uniqueness of simple demographics in the us population. *LIDAP-WP4, 2000*, (2000). [Cited on page iii.]

Taelman, R.; Van Herwegen, J.; Vander Sande, M.; and Verborgh, R., 2018. Comunica: a modular sparql query engine for the web. In *Proceedings of the 17th International Semantic Web Conference*. https://comunica.github.io/Article-ISWC2018-Resource/. [Cited on page ii.]

Taelman, R. and Verborgh, R., 2022. A prospective analysis of security vulnerabilities within link traversal-based query processing. In *Proceedings of the 6th Workshop on Storing, Querying and Benchmarking Knowledge Graphs*, vol. 3279 of *CEUR Workshop Proceedings*, 65–80. https://rubensworks.github.io/article-ldtraversal-security-short/. [Cited on page iii.]

Verborgh, R., 2021. Reflections of knowledge. https://ruben.verborgh.org/blog/2021/12/23/reflections-of-knowledge/. [Cited on page i.]

Verborgh, R. and De Roo, J., 2015. Drawing conclusions from linked data on the web: The eye reasoner. *IEEE Software*, 32, 3 (2015), 23–27. [Cited on page ii.]

Yao, A. C.-C., 1986. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, 162–167. doi:10.1109/SFCS.1986.25. [Cited on pages i and ii.]