AFRL-RI-RS-TR-2021-037

FRAMEWORK FOR INFORMATION DISCLOSURE WITH ETHICAL SECURITY (FIDES)

GALOIS, INC.

FEBRUARY 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-037 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
FRANCES A. ROSE
Work Unit Manager

/ S /
GREGORY J. HADYNSKI
Assistant Technical Assistant,
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
FEBRUARY 2021	FINAL TECHNICAL REPORT	JAN 2018 – MAY 2020

4. TITLE AND SUBTITLE

FRAMEWORK FOR INFORMATION DISCLOSURE WITH ETHICAL SECURITY (FIDES)

5a. CONTRACT NUMBER
FA8750-18-C-0051

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

Dr. David Archer
Jef Bell

5d. PROJECT NUMBER
DHSF

5e. TASK NUMBER
ID

5f. WORK UNIT NUMBER
ES

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Galois, Inc.
421 SW 6th Ave Ste 300
Portland OR 97204

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITE
525 Brooks Road
Rome NY 13441-4505

Department of Homeland Security
1100 Vermont Ave Ste 300
Portland OR 97204

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2021-037

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA# AFRL-2020-0560
Date Cleared: 18 Feb 2021

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project, we explored the feasibility of providing confidentiality protections to sensitive data while the data is being computed on, through use of client-side secure enclave technology. We designed and built a prototype system that provides query capabilities to datasets that remain cryptographically protected from inspection of any kind. Query results are limited by access control rules provided by the dataset owner. We evaluated the constructed system's performance and capabilities.

15. SUBJECT TERMS

Secure data analysis, privacy preserving technology

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**FRANCES A. ROSE**
U	U	U	UU	17	19b. TELEPHONE NUMBER (*Include area code*) **N/A**

TABLE OF CONTENTS

LIST OF FIGURES

1.0 SUMMARY

Emerging threats to protecting sensitive data have made it untenable to rely solely on confidentiality protections of data at rest and data in transit. The FIDES project aims to provide assurance that sensitive data can also be protected *during computation*. The FIDES system uses a specific privacy-preserving technology, *enclave encryption*, where specialized computer hardware available in modern processors provides a wall of encryption protection so that no other resources on the computer can access the data while it is decrypted and computed on. Using this technology, the FIDES system allows a dataset user to query an *encrypted* dataset while enforcing access controls specified by the dataset owner, limiting the query results revealed to the dataset user. The dataset owner specifies access control rules with the FIDRIS component, and the FIDO component enables creation of user-specific attributes. The FIDURA component reconciles the rules and attributes to determine if the dataset should be made available to the user, and if so, to provide database-enforceable access controls. In this project, we designed and prototyped the FIDES system, and evaluated its capabilities and performance.

2.0 INTRODUCTION

Electronic storage of data has become highly cost effective in the last two decades. This cost trend has resulted in two significant changes in organizational (and personal) behavior. First, organizations now typically store data "forever". Second, organizations collect increasingly more data to support either current or potential future analytic needs.

Both of these changes give rise to an increasing level of data sharing, either because such data can be monetized if shared, or because the existence of such long-term data provides significant opportunity for new analyses that leverage such data to offer insights unavailable in the past. For example; network traffic data is easily gathered; can be stored *en masse* for the long term; and is often shared by the network ISPs that gather the data with researchers who analyze the data to detect new trends in cyber threats.

Unfortunately, the rush to collect, store, and share large volumes of data has resulted in overlooking the sensitivity of such data and how that sensitive data should be stewarded. Returning to the example of ISPs gathering and sharing network traffic data, such data often contains personally identifiable information of network users that may be highly sensitive. HTTP request packets specify URLs that users access; HTTP request parameters include the details of user interaction with those URLs; and some network traffic includes sensitive credentials such as credit card numbers, passwords, or SSNs.

In the past, ethical stewardship of sensitive data collected by such organizations focused on strong protections of confidentiality of the data at rest and in transit. In that setting, data sharing was permitted based on the trust of the data provider in the assurances of such protections being adequate on the servers of the data consumer. However, emerging threats make it untenable to rely on such assurances of protection of data at rest and in transit. In addition, other emerging threats make it necessary to provide assurance that sensitive data is also protected *during computation.* These emerging threats include

- the rise of significant *insider threats* — trusted individuals who break that trust and exfiltrate sensitive data
- a substantial and exponentially increasing set of *network threats* — individuals or groups skilled at exploiting vulnerabilities in services that connect servers to the world-wide network and using those vulnerabilities to penetrate systems and exfiltrate data
- the relatively recent rise of persistent threats that reside on systems and can examine the content of system memory, thus bypassing data-at-rest and data-in-transit protections

These novel threats make traditional approaches to information assurance such as encryption at rest and encryption in transit significantly less effective at protecting sensitive data. In response, new *privacy-preserving technologies* are slowly emerging from the computer science research community. Among these are technologies that keep data encrypted *during computation* - that is, throughout the residence of data in main memory. The addition of these technologies to the information assurance arsenal, when properly applied, assures this *last mile* of data confidentiality and integrity. However, these technologies remain largely unexplored in terms of performance and ease of use in practical settings.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Approach

FIDES explores the performance and practical use of privacy-preserving technologies in the setting of network traffic data sharing among ISPs that collect such data, and researchers who "borrow" and analyze that data in their work. FIDES explores a specific privacy-preserving technology: *enclave encryption*, where specialized computer hardware available in modern processors assures a wall of encryption protection so that no other resources on a computer can access the data while it is decrypted and then computed on. FIDES provides end-to-end cryptographic assurance for such data: data is encrypted before it leaves the trust zone of the data provider; remains encrypted (or protected by specialized hardware that provides an encryption "wall") throughout its lifetime, including during computation; and is then unambiguously and provably deleted so that it can never be recovered after analysis is complete.

Of particular interest in FIDES are the following research questions so far unexplored in using such technology:

- To what extent is it practical to build and deploy a full ecosystem around these technologies that is easy to use for data providers, administrators, and researchers who are not experts in cryptography?
- How does performance of these technologies scale when tasked with handling realistic query loads over realistic dataset sizes?

We answer these questions by demonstrating a working prototype of this full end-to-end information assurance solution on data sets of practical size, and characterizing delivered performance of our prototype on a collection of queries over that data.

3.2 Methods

We followed the methodology below in answering these questions. Our work consisted of twelve major tasks:

Task 1 Construct basic FIDES enclave

Develop an SGX enclave capable of decrypting and processing provider's data. Implement a simple data access demonstration, without technical controls, generalized query processing, or attestation. Build initial API to provide access to enclave functionality.

Task 2 Design initial technical control language

Determine the technical controls needed, informed by the target use case and additional anticipated access control needs for IMPACT. Design a language sufficient for specifying these controls. Implement a parser and syntax checker, and an AST traverser that transforms abstract syntax into a machine-interpretable serialization for communication to the FIDES enclave.

Task 3 Design initial query language, schema definition and processing approach

Determine schema requirements and query functionality required to support target use case and additional anticipated needs for IMPACT. Design a language sufficient for specifying schema and queries. Outline approach for rewriting and processing queries. Implement a parser, syntax checker, and type checker.

Task 4 Define security architecture of FIDES and provide formal security proof.

Informed by discoveries made in the above tasks, create a security architecture specification for FIDES, document adversary assumptions, and construct a formal security proof that shows FIDES is resilient to the specified adversary types.

Task 5 Develop attestation capability

Construct a provider-side agent that interacts with the client-side enclave to perform attestation on code and static data in the enclave. Encrypt provider data and implement a means of securely transporting the key and encrypted data to the FIDES enclave.

Task 6 Implement query processing

Enhance the basic FIDES enclave to execute queries specified in the query language, without technical control enforcement.

Task 7 Implement query rewriting for technical control enforcement

Enhance query processing to rewrite queries to enforce technical controls as specified in the technical controls language.

Task 8 Integrate and exercise initial end-to-end system on target test suite

Construct a test suite for the target use case. Integrate the above components. Enhance enclave API as required. Exercise the integrated system on the test suite.

Task 9 Refine architecture and security proof

As needed, refine the architecture and security proof defined in task T4 based on what is learned building the initial end-to-end FIDES system.

Task 10 Refine query and technical control languages

Refine and extend the query and technical control languages based on what is learned building the initial end-to-end FIDES system. Explore optimizing technical control enforcement by rewriting ASTs.

Task 11 Refine query rewriting and processing

Refine and extend query rewriting and processing to support the updated languages created in task T10.

Task 12 Integrate and exercise refined system on comprehensive test suite

Integrate refined components. Design a set of tests that exercise all supported query operations on a variety of data sets, run these, and verify test results against running those same tests on plain-text data sets.

4.0 RESULTS AND DISCUSSIONS

4.1 The FIDES System

The FIDES project achieved all planned outcomes, including:

- Creation of a client-side *secure enclave* based on Intel SGX technology that
    - interacts with a data catalog to identify data sets of interest and the data provider servers where that data can be obtained
    - interacts with a data provider's server to securely prove the enclave's authenticity and integrity
    - accepts encrypted data from the provider's server
    - provides a relational database and associated query capability to the user for analyzing the data *while it remains cryptographically protected from inspection of any kind*
    - provides an access control enforcement point that limits the query results revealed to the user
    - and assures that at an agreed-on time, the data becomes entirely inaccessible any time in the future, thus preventing future data theft.
- Creation of the server-side data mediation manager that interoperates with the client-side enclave to perform the authentication, integrity assurance, and data provisioning described above
- Creation of a means to specify access control rules (the FIDRIS component) that are reconciled (by the FIDURA component) with the user's attributes (specified via the FIDO component) to determine if a dataset should be made available to the user, and if so, to provide enclave-enforceable access controls
- Creation of a data catalog and browser capability that interacts with both of the above software agents, thus allowing data providers to advertise data sets, users to browse and select relevant data sets, and connecting providers and users to enable secure sharing as described above
- Porting of the FIDES prototype to a second operating system to demonstrate flexibility of the implementation
- Analysis of performance of representative queries on synthetic datasets representative of real network traffic data set size and schema

Figure 1 illustrates the architecture of FIDES. At left in the figure is the Data Mediation Manager (DMM). The DMM is a software agent running on a server that stores potentially sensitive information and provides it under certain conditions to researchers who the provider allows to analyze the data, but not inspect it directly. The DMM is a trusted zone for storing the data throughout its normal lifetime. We do not address the security of the DMM server — that topic is outside the scope of our research.

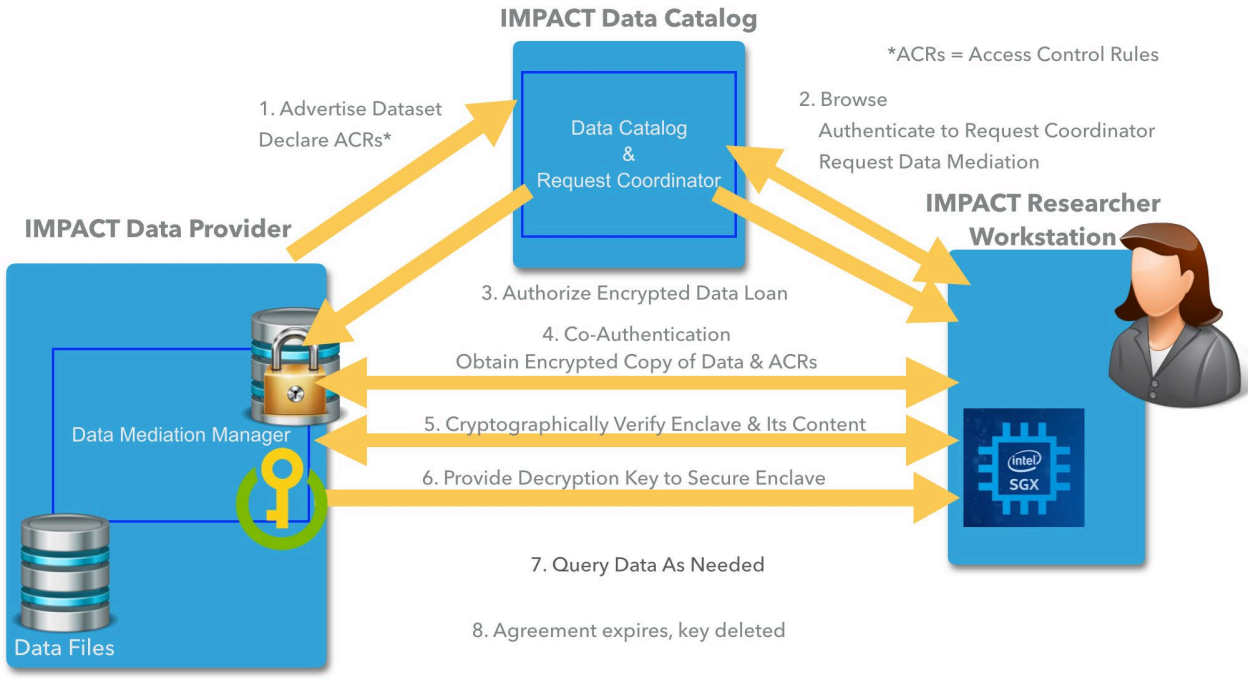


Figure 1: FIDES Architecture

Sharing of a sensitive dataset begins when the DMM chooses to advertise the dataset's existence. The DMM does this by providing a description of the dataset and corresponding access control rules. This information is provided to the Data Catalog and Request Coordinator subsystem (DC), shown at top in the figure. The description may include a high-level textual explanation of the meaning and intended uses of the data. The description also includes the *schema* of the data: what fields are included, the data types in those fields, and associated metadata such as the units or representations used for those data types. In FIDES, we assume that such schema information may be sensitive, but that access to that schema is controlled by the policy that administrators use to grant access to these data advertisements (which we call the *data catalog*). The access control rules (ACRs) that apply to a dataset may be conditional on the *attributes* of the researcher who wishes to analyze the data. For example, one attribute of interest may be whether or not the researcher is a US person, as defined by the US Department of State. Thus a data provider may equip the same dataset with one or more sets of ACRs, allowing different kinds of results to be made available to diverse researchers depending on their attributes. The FIDURA component determines which datasets are available to a researcher based on their attributes, and which ACRs to enforce. FIDES provides a language for defining ACRs via the FIDRIS component, and provides the FIDO component for users to browse datasets and specify user attributes.

At any time, a user (which we call here a *researcher* to avoid ambiguity) who has an approved account may browse the DC. Such users are shown at right in the figure above. Each user can see datasets and associated ACRs that may apply to them, as determined by FIDURA. Once such a dataset is found, the user may request access via the DC. In response, the DC securely provides

the researcher with a unique key, or *access code*, along with the URL of the relevant DMM. The DC also provides this access code securely to the relevant DMM, thus facilitating communication between the DMM and the researcher's *secure encrypted computation enclave* (SEC) about the dataset of interest.

Once the researcher has selected a dataset with associated ACR set and obtained an access code, the researcher's SEC initiates access to the data. First, the SEC deploys an Intel SGX enclave.

The deployed enclave contains the software subsystems shown in Figure 2. Included are the necessary functionality to prove its authenticity, and the content of its included software, to a DMM. The enclave also includes software to allow the user to ask queries, a relational database to process those queries, a result access control subsystem to apply the relevant ACRs to limit what results the user will be allowed to see, and a data decryption engine to decrypt the dataset internally to the enclave and install it in the relational database so that it can be queried.

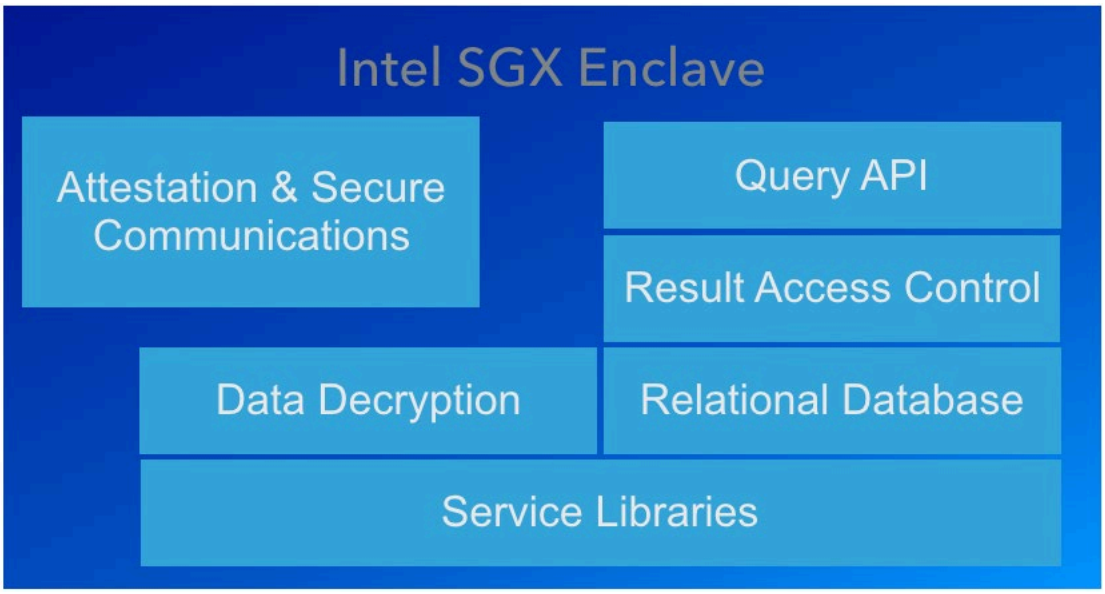


Figure 2: An SGX Enclave

Upon deployment, the user provides the enclave with the access code and URL provided by the DC. The enclave then contacts the relevant DMM (guided by the URL) and begins the authentication process. First, the enclave proves that it is a genuine SGX enclave. This proof is discharged by use of the built-in *enclave signing* capability in Intel's SGX architecture. The DMM receives information digitally signed by a key held in the enclave's processor logic, and checks this signature with an Intel authentication server at a fixed domain name on the Internet. (Proving that this service is not falsifiable is beyond the scope of this project.)

Next, the enclave proves to the DMM that its software content is exactly the same, byte for byte, as the expected content of a FIDES enclave, which is a public value. This proof is discharged by use of the Intel SGX *attestation* feature.

Once authenticated and attested, data is provisioned to the enclave. The dataset and associated ACRs are encrypted by the DMM with a fresh AES-256 key. Once encrypted, this file may be transferred to the workstation containing the enclave with no further security. The key to the file is then provisioned directly to the enclave via a secure TLS channel. The workstation and researcher never obtain access to this key. Thus the enclave is the only agent outside the DMM's trust zone that can decrypt the data. A major feature of the Intel SGX architecture is that no process, not even the operating system, running on an SGX-equipped processor can "see inside" the enclave's boundary. Thus even though the dataset and ACRs are decrypted within the enclave, *only the code inside the enclave can access either part* of the provisioned data.

Once the enclave has decrypted the data, it loads the data into its internal relational (SQLite) database. This database is contained entirely within the enclave, so no other access to this data is possible from outside the enclave boundary. Similarly, the ACRs are decrypted and loaded into the Result Access Control subsystem, and cannot be accessed or modified by any agent outside the enclave. Thus the confidentiality of the data and the integrity of the ACRs are ensured by the enclave mechanism.

Once data and ACRs are ready, the enclave offers a query API to the researcher. Queries are formed in normal SQL, the most popular database query language, providing users with a rich capability for querying data. Each query is processed by the internal SQL database. Then, the ACRs are applied to the query result. Results allowed by the ACRs are then provided to the researcher. Thus queries of the data proceed, and (controlled) results are provided, even though the researcher has no direct access to the data.

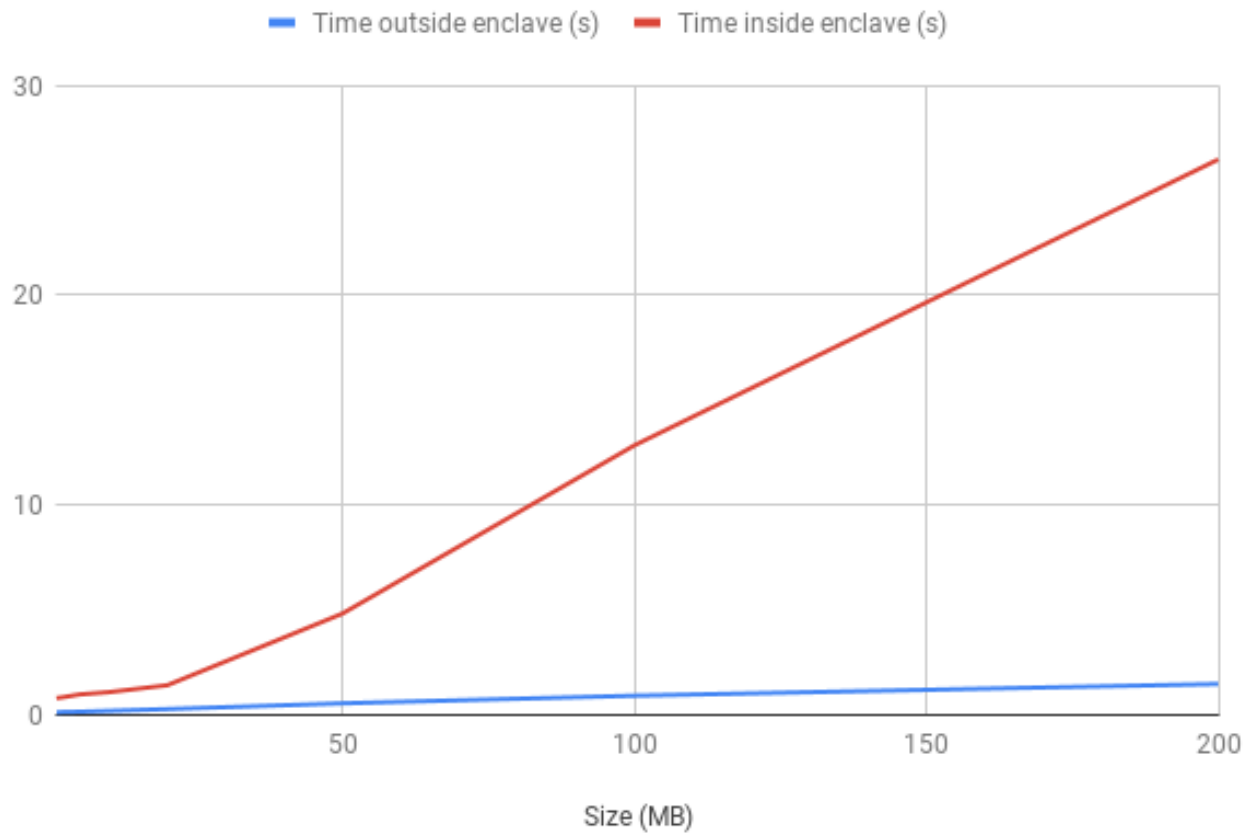
Querying may continue in this way for as long as the ACRs allow. Our ACR language includes the notion of *expiry* that can limit this time period. Once the expiry date is reached, the enclave simply clears all decrypted data and erases the provisioned key. At that point, the dataset is no longer accessible for any purpose by the researcher.

4.2 Performance of FIDES

We synthesized data sets of various sizes, and constructed a set of representative queries to run on those datasets. We then measured query response times of FIDES and compared them to response times for the same relational database running the same queries over the same data *outside* the enclave environment, but on the same processor. The graph in Figure 3 illustrates the average query response time (in seconds, on the vertical axis) for our test queries as a function of the size of the synthesized dataset (in megaBytes, on the horizontal axis). Our measurements include the time it takes to load (and in the case of the SGX enclave, decrypt) the data into the database. Timed results for the privacy-protected (SGX enclave) database are shown in red, while results for the unprotected database are shown in blue. We verified that the answers to both sets of queries matched completely, using a set of ACRs that allowed full access to the data.

As shown in the graph, both implementations show a (roughly) linear increase in access latency as dataset size increases. Note that this behavior is expected, because our measurements include the time to load (and in the case of SGX, decrypt) the data. However, the privacy-protected database latency demonstrates a much higher dependence on dataset size than the unprotected

database. We expect this penalty will be much lower when the load/decrypt cost is amortized



over many more queries.

Figure 3: FIDES Performance

4.3 Evaluation of FIDES Access Control Policies

We developed a diverse set of ACRs to check whether our access control enforcement point in the enclave functioned as expected. We ran the same queries as before, and manually inspected all query results. We confirmed that all provided access modes allowed by our access control language operated correctly on each query. We concluded that the access control enforcement logic operated as expected.

5.0 CONCLUSIONS

In this project, we successfully demonstrated the feasibility of protecting the confidentiality of sensitive data while the data is being computed on. We constructed a prototype FIDES system, using client-side enclave encryption to provide these protections. FIDES allows a dataset user to query an *encrypted* dataset while enforcing access control rules specified by the dataset owner, thus limiting the query results revealed to the dataset user. Performance measurements demonstrate that queries executed approximately an order of magnitude slower with FIDES than with an unprotected database, with just a linear increase in access latency as dataset sizes increase. We believe this is a reasonable performance tradeoff, given the significant confidentiality protections provided by FIDES.

6.0 List of Acronyms

ACR	Access Control Rules
AES	Advanced Encryption Standard
API	Application Programming Interface
AST	Abstract Syntax Tree
DC	Data Catalog
DMM	Data Mediation Manager
FIDES	Framework for Information Disclosure with Ethical Security
FIDURA	FIDes Utility and Risk Assistant
FIDRIS	FIDes RISk assessor
FIDO	not an acronym
HTTP	HyperText Transfer Protocol
IMPACT	Information Marketplace for Policy and Analysis of Cyber-risk and Threats
ISP	Internet Service Provider
SGX	Software Guard Extensions
SEC	Secure Encrypted Computation Enclave
SSN	Social Security Number
SQL	Structured Query Language
TLS	Transport Layer Security
URL	Uniform Resource Locator
US	United States