Federal Register Notice 87 FR 35250, https://www.federalregister.gov/documents/2022/06/09/2022-12432/request-for-information-on-advancing-privacy-enhancing-technologies, June 9, 2022

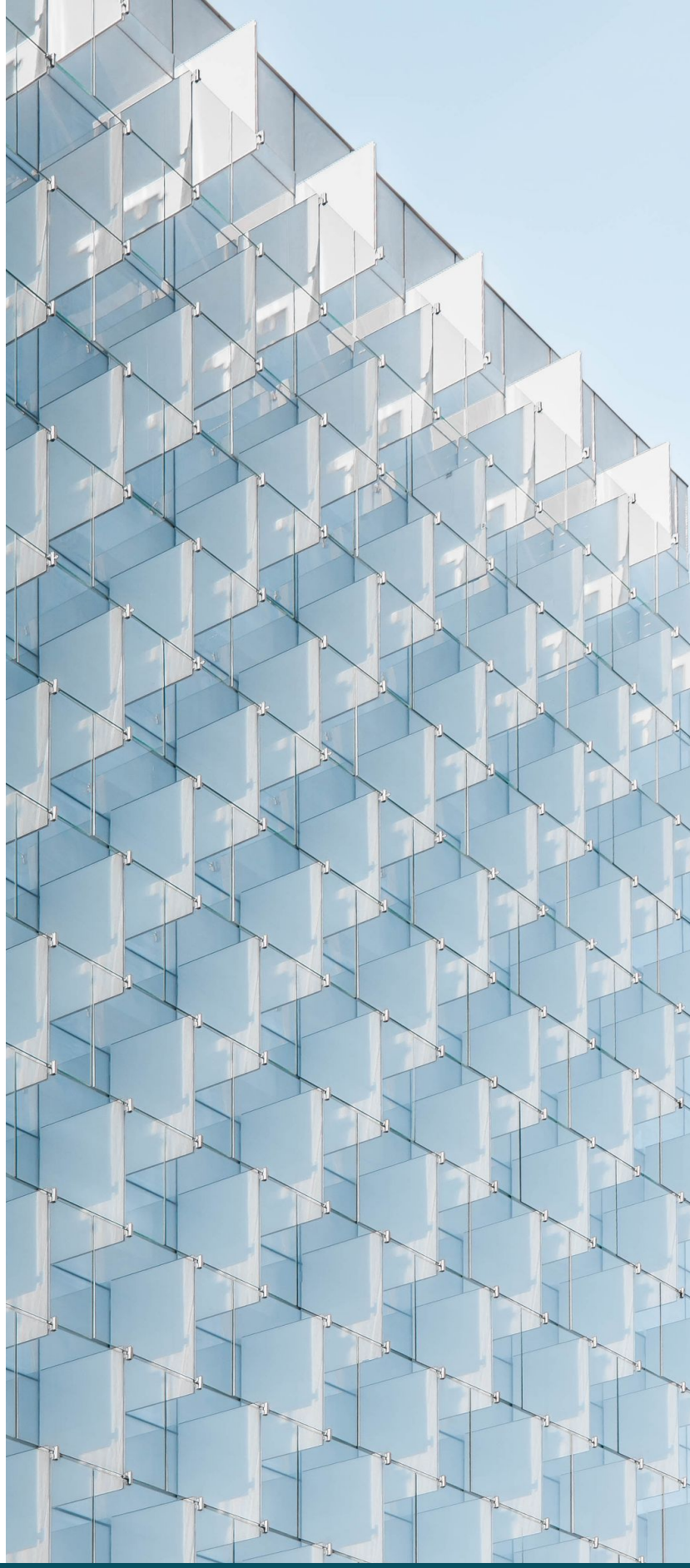
Request for Information (RFI) on

Advancing Privacy Enhancing Technologies

Agita Labs

AGITA LABS

REQUEST FOR INFORMATION RESPONSE



JULY 8, 2022

DRIVING DISCOVERY | PROTECTING PRIVACY

Table of Contents

Cover Page

AGITA LABS

To whom this may concern:

At Agita Labs, we do not try to stop hacking.. Instead, our technology protects data *despite ongoing hacking*. This change in mindset is spearheaded by the work of Dr. Todd Austin and Dr. Valeria Bertacco, two computer science professors from the University of Michigan. In 2017, Austin and Bertacco built Morpheus, a secure computing platform that was designed to protect data from hacking. DARPA, the research arm of the US Department of Defense, put this system into a commercial red-teaming effort for three months, during which 500+ cybersecurity researchers were unable to successfully attack the system even a single time.

At the heart of Agita Labs' solution is a novel patented technology called *sequestered encryption* (or SE, for short). Sequestered encryption builds a hardware-based cryptographic wall between all software and sensitive data. As SE-protected software runs, all sensitive data is encrypted all the time, even when it is processed and analyzed. In fact, no one, not even programmers or superuser IT staff, can see data protected by sequestered encryption except the data owner who originally encrypted the data.

Agita Labs is excited to be considered for the development of the national strategy on privacy-preserving data sharing and analytics to better benefit individuals and the society as a whole. We know that secure computation holds great promise for advancing data breach protection, privacy enforcement, and zero-trust data sharing.

The opportunity to solve problems and create change around privacy for a range of institutions is the most attractive aspect of what we do. In this response, the objectives are to demonstrate what our technology can do, some barriers we have had with adoption when commercializing the technology, as well as some barriers around the technology itself.

Best regards,
Todd Austin, CEO
Valeria Bertacco, Chief Scientist
Sara McLean, Head of Business Development
Agita Labs, Inc., Ann Arbor, MI

Executive Summary

Secure computation holds great promise for advancing data breach protection, privacy enforcement, and zero-trust data sharing. This promise lies in the cryptographic-strength defenses that secure computation provides for data confidentiality – defenses that persist even when the system's software has been penetrated by attackers. These powerful defenses have emerged in several secure computation approaches: homomorphic encryption (HE) and multi-party computation (MPC). In addition to stopping data breaches, these novel secure computation technologies also enable new forms of zero-trust data sharing, making it possible to share sensitive data with untrusted parties while retaining control over who can view, process and learn from their data.

Existing secure computation technologies are not without their drawbacks, however. Today's secure computation frameworks possess many barriers to adoption: limited security defenses, performance overheads, and programming challenges. These constitute significant barriers to their adoption, limiting the organizations that will deploy these technologies and the applications for which these technologies can be employed. Additionally, commercial challenges exist in deploying secure computation technologies, including challenges surrounding consumer education, technology validation, market identification, standardization, and funding.

Agita Labs' has developed a novel form of hardware-based secure computation, called sequestered encryption (SE). Our SE technology addresses many of the technical concerns that have plagued other secure computation platforms. The capabilities of SE outmatch existing secure computation frameworks, providing confidentiality and integrity checking capabilities, plus safe data releases. The performance of SE-based secure computation is orders of magnitude faster than existing competing secure computation technologies, and with additional optimization and integration into the CPU processor, these overheads could be further reduced. Finally, the programmability of SE is much more approachable to developers, requiring only a few hours of training, rather than the highly idiosyncratic approaches embodied in competing approaches.

In this RFI response, we advocate for the use of sequestered encryption as an emerging technology that can address many of the technical barriers currently hampering the adoption of secure computation defenses. Moreover, we articulate with several recommendations for the US government to aid in overcoming both the technical and commercial barriers to the wide-spread adoption of secure computation technologies. These recommendations include support for additional funding for secure computation research and commercialization, exploration of a broader palette of secure computation technologies, and prioritizing government deployment of secure computation in its own sensitive IT operations.

The Promise of Secure Computation

Why Today's Data Security Is Not Very Secure

Data breaches are a regular––expensive and embarrassing––part of the current tech industry. IBM estimates the average data breach cost at more than $4M, not including damage to public reputation and client relations. The root of this problem lies in popular approaches to data protection. Virtually all systems (even the most advanced systems that utilize trusted execution environments like Intel's advanced Software Guard Extension) protect data with software. As a result, to protect data we must prevent software hacking––a task that remains an eternally unsolved challenge in the security community.

Software hacking occurs when clever attackers identify vulnerabilities (typically in the form of bugs) in software that allow sinister exploits to step around the security defenses deployed on systems. Nearly all security breaches in the news are the result of software hacking, so stopping these exploits is a prime mission of the security industry. A closely related, but equally important, form of hacking is side channels. Side channels occur in the observable characteristics (e.g., run time) of software and hardware, revealing sensitive data being processed inside an application. While side-channel attacks are mostly academic today, there have been a few that have risen to the critical concerns, e.g., Kocher's attack that exposed RSA keys remotely, and most advanced data security technologies will work to address software hacking and side-channel attacks. In summary, if you cannot stop software hacking and side channels, you cannot protect data from attackers.

Why Not Put a Stop to Hacking?

A vast majority of the security industry is focused on stopping and detecting software hacking. To stop software hacking, security professionals attempt to identify and fix all the "bugs" in a program. Bugs are programmer errors, which often do not break an application, but attackers find clever ways to exploit these bugs. When bugs are found, they need to be fixed quickly before attackers can exploit them. This approach to security is often called "Patch and Pray" because finding *all* bugs is essentially impossible. If you ask a programmer if there exists any software without bugs, they will most likely say "No." Software is simply too complex, too rapidly evolving, and too intractable to lend itself to any form of high precision "bug hunting. "Clearly, the odds of a software hack favor the attacker.

An important and powerful tool for bug hunting for formal verification. With formal verification, mathematical methods are used to prove that a program doesn't contain bugs that attackers could exploit. Formal verification is incredibly powerful because, when it works, it can find bugs before they can be exploited by attackers. The challenge with formal verification is two-fold; first, real programs are usually too complicated to fully analyze with formal verification. Formal security verification is a process of proving that something cannot happen, which is one of the most difficult computational challenges in computer science. Second, formal verification cannot identify the exploits that attackers will invent in the future. These exploits, which are often called zero-day exploits, are a primary concern of the security community, because these bugs represent those which the attacker community is aware of and the security defense community is not!

Further, even if organizations could find and fix all software bugs, they would still be susceptible to side channel attacks, which simply observe the operation of software and hardware to infer their secrets. Sophisticated and well-meaning developers can easily write completely bug-free code that is riddled with side channels, allowing any listening attacker to quickly understand the secrets held within the software. All considered, it is important to recognize that mainstream security defenses today cannot durably stop data breached. Vigilant and dedicated security teams can make breaches much harder, but these organizations are still at risk of getting breached.

Secure Computation Address Key Technical Challenges

Secure computation is a new form of computer computation that works directly on encrypted data without software needing to hold a data access key. **Examples of systems that support this capability are those based on homomorphic encryption (HE) and multi-party computation (MPC). Systems that support secure computation build a cryptographic wall between all software and data, which immunizes these systems against disclosures of sensitive data via software hacking. While software hacking is effective on systems that perform computation in the clear, it has little agency over secure computation because once the hacker penetrates the system, all there is to steal is encrypted data. As illustrated in Figure 1, this desirable property of secure computation is referred to as *zero software trust*, because secure computation can maintain data security even if all the software in the system is hacked.**

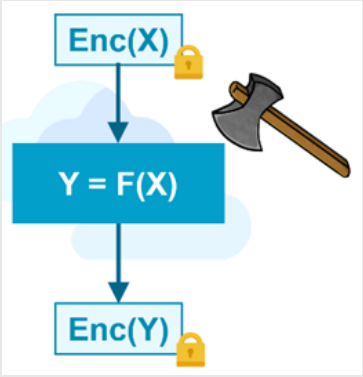


Figure 1: Secure computation runs directly on encrypted data.

Secure computation can also be a durable defense against side-channel attacks, if the underlying implementation of the secure computation capabilities doesn't leak information through observable channels, e.g., memory or operational timing. Most secure computation frameworks strive to eliminate vulnerabilities due to software hacking and side-channel attacks.

Secure Computation Addresses Privacy Challenges

Secure computation has significant potential future value to computing hardware that incorporates these capabilities, due to its ability to durably stop data breaches and create new forms of safe data sharing. The approach doesn't stop software hacking (which is a much more difficult and perhaps impossible task), but rather it renders the data breach inert since attackers can only steal encrypted sensitive data.

Zero software trust is more valuable than simply being a powerful security measure, it also creates many high-value opportunities for zero-trust data sharing. **With zero-trust data sharing, it is possible to process sensitive encrypted third-party data without risk of attackers or programmers seeing, abusing, or stealing the data. For example, with secure computation, one could send their encrypted genome to an online (and potentially untrusted) server, which would then run a proprietary disease profiling algorithm on it using secure computation. The results of this computation would not be visible to anyone except the original data owner that encrypted the data, thus guaranteeing the privacy of the genome data. The prospect of zero-trust data sharing creates many business opportunities for secure computation in medical data sharing, fintech data sharing, genomics research, advertising analytics, private smart blockchains, secure health monitoring, secure avionics, privatized surveillance, *etc*.**

Technical Barriers to Adoption of Secure Computation

While the promise of secure computation is great, the technical challenges in deploying it are also equally great. These challenges are both commercial and technical in nature, and in this section, we focus on the technical challenges, after which we illuminate the commercial challenges that we have experienced in our quest to deploy secure computation in customer operations. In this section, we limit our discussion to the technical barriers associated with homomorphic encryption (HE) and multiparty computation (MPC), the two dominant secure computation technologies in the marketplace today.

Performance Barriers: **HE and MPC technologies are notoriously slow, with HE suffering from significant computation overheads and MPC suffering from heavy communication overheads. Compared to unprotected algorithms, these technologies can slow program performance by 100x – 1,000,000x slower. While there are certainly applications that can tolerate these slowdowns, there are many other important application areas that cannot, such as machine learning training, computer vision applications, and real-time applications.**

Limited Defense Capabilities: **HE and MPC technologies also suffer from limited security capabilities. Today, these technologies focus on the confidentiality of data and lack any support for ensuring the integrity or availability of computation resources, which are two other critical security concerns. For example, if a voting machine were built with HE technology, it could certainly keep votes secure from disclosure, but today's HE frameworks couldn't tell if an attacker manipulated the HE secure computation to add all the votes to their own preferred candidate. These types of attacks are called integrity attacks, because the attack compromises the integrity of the secure computation, making it possible for the attacker to create a new algorithm that likely benefits the attacker. The MPC community has begun to address this concern with zero-knowledge (ZK) proofs, but these proofs are *i)* specific to an algorithm and often require PhD-level expertise to craft, and *ii)* require significant additional computation resources to verify computation integrity. Finally, without strong integrity checking mechanisms existing secure computation frameworks cannot perform safe disclosures of encrypted data. A safe disclosure is a capability that allows the decryption of a specific program value that the data owners agree to release. Without strong integrity checking, attackers could abuse a decryption capability, and thus, today's secure computation frameworks require that the original data owner decrypt any value to be released. While a safe approach, this creates many application-specific deployment challenges. For example, regulatory access to secure computation isn't possible without the original data-owner performing the decryption. This would be a problematic situation especially if the data owner was committing infractions that the regulatory agency was trying to detect.**

Significant Programming Challenges: **The nature of HE and MPC defenses is that they don't protect software directly, instead, they protect linear integer arithmetic. As such, programmers must express their algorithms in the form of a linear integer mathematical expression before it can be protected by these frameworks. This requirement creates significant programming challenges that make it difficult or impossible to express certain algorithms. For example, the "less than" inequality test poses a significant challenge to HE, due to its non-linear nature. Thus, one doesn't see applications that require sorting being protected with HE defenses, since sorting algorithms rely heavily on relational tests. As such, mission critical applications such as databases cannot be readily protected with HE technology. In addition, the countless applications that rely on strings and floating-point numerics have significant challenges with HE adoption, since these fundamental data types are not well supported in today's secure computation frameworks.**

Commercial Challenges to the Adoption of Secure Computation

With any startup comes multiple challenges in regards to adoption and commercialization. With a deep tech startup such as Agita Labs or other privacy enhancing technology companies, the challenges are often more complex. Being able to solve a problem for any opportunities that a privacy enhancing technology can present from a practical as well as economic standpoint can be the key challenge.

Education: **Educating the market on how to use a PET and the benefits from it has been the single largest challenge in terms of adoption. Providing more training and development such as workshops, seminars, and education around what a privacy enhancing technology provides is needed for a successful commercialization. The need for more education around privacy enhancing technology is the largest challenge the PET community faces in terms of adoption.**

Validation: **Small or large enterprise businesses want to make sure that anyone they are working with has accreditation and third party validations or frameworks in place. Companies want to make sure when they are partnering with a new vendor that certain boxes are checked before moving forward. This holds true even more when you are working with data and protecting privacy. Privacy-enhancing technology is still new, therefore causing some hesitation on privacy and security conveners for their own organization. Being a small startup business and not being able to have specific proof points or third party validation can ultimately cause an enterprise business to not move forward.**

Marketing Identification: **Despite there being many challenges to overcome with new and innovative technologies such as a PET, narrowing down the correct market to target who understands the need for this technology has been another challenge. Once the defined markets have a better education and understanding on how a PET can be utilized in their environment, standardization will come soon after.**

Standardization: **Enterprise businesses do not always grasp the need for adopting a new technology within their organization. From the research and multiple conversions that have taken place, companies know very little around PETs. New technology is always a disrupter in the short run. The enterprise businesses have to upgrade their existing procedures and systems which can cause a major disruption. Large enterprise businesses have a lack of knowledge on how much time/resources a PET will need with their internal teams and until using a PET becomes a more standard practice, adoption to commercialize a PET will be at a slower pace.**

Funding: **Investors are always extremely excited and eager to hear about what Agita Labs and other privacy enhancing technology companies (and researchers) are working on, but shy away because they themselves are unclear as to the growth capacity of a PET, and what specific need or problem will be solved. Because a PET is still relatively new to the market, finding partners and channels to adopt the technology has been difficult.**

Ways the Government Could Lessen Commercial Barriers

We see several opportunities where government involvement and resources could speed the adoption of secure computation, by lessening technical and commercial barriers.

Secure Computation Research Funding: **Programs such as DARPA's SSITH and DPRIVE programs have been vital in advancing secure computation capabilities. DARPA SSITH's program led to the sequestered encryption technology detailed in this RFI. DARPA's DPRIVE program is developing significant improvements in the performance of homomorphic encryption, through the deployment of specialized hardware accelerators. Additional funding directed specifically toward secure computation (or privacy-enhanced technology) research would be a welcome opportunity, for both secure computation researchers and startups.**

Broadening the Pallet of Secure Computation Capabilities: **Today, much of the focus in the secure computation world is on homomorphic encryption. This is certainly warranted, since homomorphic encryption stands alone in its ability to minimize trust in the system software and hardware. But as the need to overcome the technical barriers to HE adoption persist, we feel that the government has become aware that additional more capable secure computation solutions do exist, such as Agita Labs' sequestered encryption.**

Prioritize Government Deployment of Secure Computation Capabilities: **Government entities see the value and problems that a privacy enhancing technology can provide. With that, if government entities start utilizing a PET this will help with adoption on the commercial side as well. Researchers and startups in the secure computation arena would welcome the US government in taking a leadership position in adopting secure computation technologies in its own IT operations, to advance its ability to stop data breaches, promote zero-trust data sharing capabilities, and provide additional commercialization opportunities for emerging secure computation products. On top of this, requiring new specific requirements or policies on data and privacy would help generate growth for privacy enhancing companies to work together to solve a problem that hasn't fully been solved as of yet.**

Commercialization Grants: Government grants for small start ups would be extremely helpful for multiple reasons. A grant can help with the business expansion to develop new marketing strategies, improve business production and assist with overall business growth. With the help of grants from the government in terms of a privacy enhancing technology this can also help with technology adoption as well as training and development. Along with the obvious benefits a grant can provide when trying to commercialize privacy enhancing technologies, grants can improve project outcomes and expansion. These grants can accelerate a small start-ups business timeline which can lead to improved outcomes for the business.

Sequestered Encryption: Under the Hood

Sequestered encryption is enforced using a small hardware component called the SE enclave. The SE enclave is a trusted hardware processing element that can expose a secret key from the data owner (using industry-standard public-key cryptography), and then operate directly on encrypted program data using simple processing commands, such as ADD, MULTIPLY, and XOR. Since the results of any computation are encrypted, the inputs and outputs of any computation remain secret. The SE enclave is deployed into the cloud using existing FPGA-class nodes in Azure and AWS or on-premises with Intel-Altera CPUs, making this powerful hardware security technology ready to use in today's security and privacy-sensitive applications.

Sequestered encryption implements *encrypted computation*, which provides a highly durable defense against the disclosure of sensitive data because it sequesters (hides) all decrypted sensitive data and the data owner's keys used to access that data inside the hardware SE enclave. No software, including software from Agita Labs or the operating system, can access information inside the software-free SE enclave. Thus, even if the system's software gets hacked, not even the hacked software can get access to sensitive decrypted data or keys. In addition, SE enclave computation is free of any control, memory and timing side channels, and thus, you can rest assured that if attackers are observing the computation, there is nothing to be learned about your sensitive data.

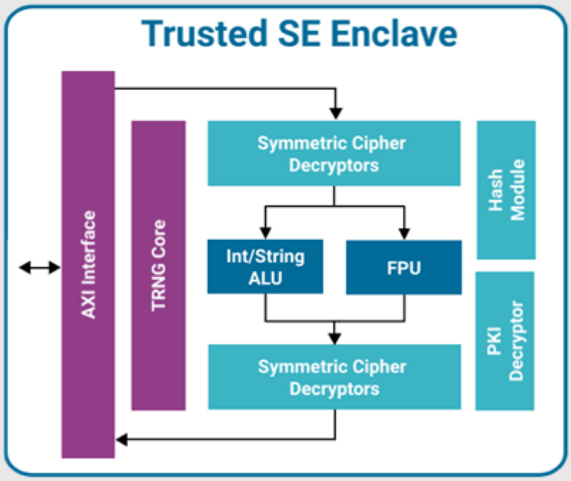


Figure 2: Agita Labs' sequestered encryption enclave.

The SE enclave enforces the integrity of any computation using a powerful patented *computational fingerprinting* technology. Computational fingerprints allow users to verify that *i)* the intended inputs to the computation were used in creating a result, and *ii)* the full unperturbed computation was run on the expected inputs. Any form of hacking to rearrange, replay, or otherwise manipulate SE computation is readily detected. This facility is invaluable in its ability to detect if the system software is being actively hacked or modified with malicious intent (e.g., supply chain attacks).

Added to this, our novel patented *safe datagrants* technology forms the basis for analysis of encrypted data, monetization of 3rd-party encrypted data, selective release of computed results, and 3rd-party auditing of encrypted data storage and computation. Safe datagrants allow a specific computation to decrypt a computed value (or transfer it to another security domain), as long as the computation that produced the value was not perturbed in any way.

Programming Interfaces

In addition to on-premises installation, Agita Labs has worked with Microsoft, Amazon, and Intel to deploy sequestered encryption into the Azure and AWS clouds. Deploying SE technology ensures that the computation is cryptographically secured to be confidential, undisturbed, and able to safely disclose data. Programmers utilize these capabilities in their business operations to:

- **Protect sensitive data from data breaches**
- **Implement zero-trust data sharing of sensitive data with potentially untrustworthy parties**
- **Process and analyze sensitive 3rd-party data that is always encrypted and confidential**

```
// calculate function F/DF value using Newton-Raphson method
enc_double rn_solver(enc_bool& converged, double maxerr,
                     unsigned maxiter, fn_type f, fn_type df)
{
  unsigned iter;
  enc_double val = 1.0;

  converged = false;
  for (iter = 0; iter < maxiter; iter++)
  {
      converged = enc_fabs(f(val)) <= maxerr;
      val = enc_cmov(converged, val, val - f(val)/df(val));
  }
  return val;
}
```

Figure 3: An example code protected with sequestered encryption, Newton-Raphson's Algorithm

Programmers access TrustForge capabilities using simple extensions to the C++, Python, or JavaScript programming languages. Programmers simply declare protected data types in their software (including integers, floating point, Booleans, and strings) and rebuild their programs to utilize SE computation; then computation on encrypted data is directed to the SE enclave where it is protected by cryptographic-strength defenses. Moreover, the SE programming interfaces enforce that programmers do not introduce vulnerabilities into their code, making SE-based programs *secure-by-construction*: if a computation compiles and runs with SE defenses, it is safe from all known forms of software hacking, data disclosure, and integrity attacks.

Competitive Analysis

The hardware integration and acceleration of the SE enclave give a significant advantage over HE and MPC frameworks, which require notably more computation and network communication. Experiments with the open-source VIP-Bench privacy benchmarks (co-developed with University of Michigan, NYU, and Addis Ababa Institute of Technology) have demonstrated that sequestered encryption technology can be many orders of magnitude faster than competing secure computation technologies. This advantage means that sequestered encryption can tackle performance-sensitive applications beyond the ability of other secure computation technologies (e.g., machine learning, recommendation, and computer vision applications).

Compared to other privacy-enhanced computation technology, Agita Labs' sequestered encryption delivers the most capable, performant, programmable, and secure computation on the market. Alternative secure computation technologies include homomorphic encryption (HE), or multi-party computation (MPC).

Significantly Better Performance: **The hardware integration and acceleration of the SE enclave give a significant advantage over HE and MPC frameworks, which require notably more computation and network communication. Experiments with the open-source VIP-Bench privacy benchmarks (co-developed with University of Michigan, NYU, and Addis Ababa Institute of Technology) have demonstrated that sequestered encryption technology can be many orders of magnitude faster than competing secure computation technologies. This advantage means that sequestered encryption can tackle performance-sensitive applications beyond the ability of other secure computation technologies (e.g., machine learning, recommendation, and computer vision applications).**

Enhanced Capabilities: **Unlike mainstream security defenses, sequestered encryption defenses are not vulnerable to any known form of software hacking or side-channel attacks. In addition to containing software, virtually all TEEs today share microarchitectural resources with untrusted software making them susceptible to control, memory, and timing side channels. Unlike HE and MPC, the SE enclave can both protect the integrity of secure computation and safely release privacy-preserving information, if allowed by the data owner. In contrast, HE and MPC frameworks typically only release information by requesting the original data owner to decrypt those results. This approach becomes problematic in regulatory auditing applications, where the data owner may choose not to share sensitive data with a regulatory agency.**

Straightforward to Program: While other secure computation frameworks protect mathematical expressions, sequestered encryption protects CPU processing commands. To protect computation with HE and MPC requires programmers to express their entire application as a mathematical expression. Moreover, HE often limits the depth of computation that is possible. These requirements negatively impede program development, requiring developers to perform potentially major surgery on an application (and brush up on their Taylor Series expansions). Sequestered encryption, in contrast, protects software directly, allowing programmers to readily port their existing unprotected applications to use SE defenses. Finally, SE-protected programs can utilize encrypted floating-point and string values, which typically pose a significant challenge to programmers in other secure computation solutions.

More Mature and Flexible Cryptography: **SE technology is built on time-tested standard asymmetric and symmetric key ciphers. This ensures that the cryptography used in the system is mature and has received significant analysis by the cryptography community. In contrast, other secure computation frameworks, such as HE and MPC, often rely on application-specific cryptographic defenses that are less mature and have not yet received significant attention by the cryptography community. Additionally, as quantum computing inches closer to the mainstream, there is growing concern about the ciphers that will be appropriate to a post-quantum world. SE technology can readily incorporate post-quantum ciphers into its defenses once the cryptography community reaches consensus on what these ciphers should be.**