Automatic Cryptography for Data Centric (ACDC) Security

POC: Dr. Nabil Schear, Senior Staff, nabil.schear@ll.mit.edu

Overview

Automatic Cryptographic Data Centric Security (ACDC) is a suite of technologies that enable the protection of data both at rest and in transit. ACDC provides access to secure computation nodes distributed throughout a network, while strictly enforcing confidentiality and integrity policies on all data globally. This global approach allows for fine-grained access control according to rich, expressive policies, and fungible secure computation.

Need

Today's security model focuses on securing massive servers, and protecting networks. This creates an environment where attackers can focus all of their effort on a small number of large, high-value targets that give massive returns with a single success. This model is responsible for many high-profile incidents, such as the 2013 Target breach [1], the 2015 OPM hack[2], and the 2016 Yahoo! breach [3].

The trend of catastrophic breaches of high-profile servers has shown no sign of slowing down, and highlights a flaw in the core design philosophy of the current state of security. As long as there are high-value targets there will be high-profile attacks against them. Securing users' data requires restructuring to a distributed, data-centric model.

[1] Krebs, Brian. "Sources: Target investigating data breach." *Krebs on Security* (2013).

[2] Bisson, David. "The OPM breach: Timeline of a hack." *Tripwire* (2015): 1-8.

[3] Thielman, Sam. "Yahoo hack: 1bn accounts compromised by biggest data breach in history." *The Guardian* 15 (2016): 2016.

Our Approach

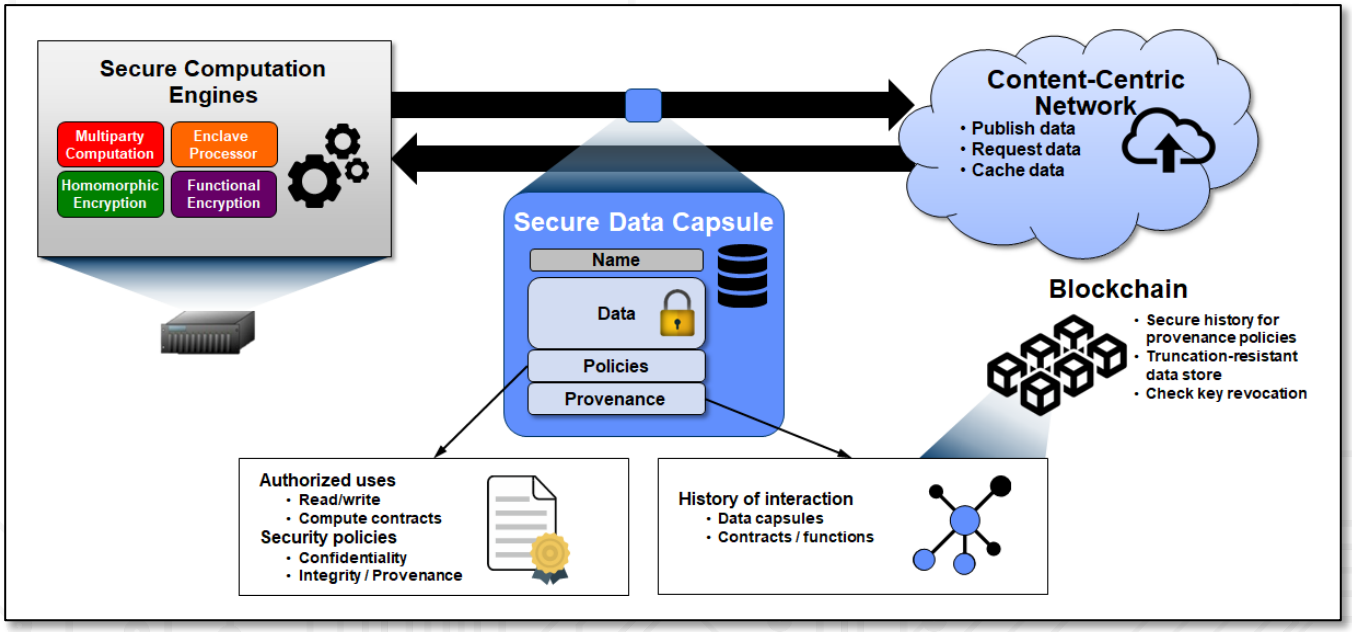
ACDC upends the current server-and-network focused approach, and instead focuses on protecting data:

- Security is fundamental priority, and all data is put into secure capsules with baked-in cryptographic protocols
- Capsules can easily be spread throughout the network and cached redundantly
- Policies can be written to ensure the confidentiality and integrity of a capsule
- Every data interaction has a permanent record of who acted on the data, and what was done to it

Benefits

- Robust policy language allows for fine-grained control of confidentiality and provenance
- Protection of data as a first class citizen
- Compatible with many different secure computation technologies, including Multi Party Computation, Functional Encryption, and Secure Enclave Computation
- Decentralized design allows for natural redundancy and resiliency
- No single points of catastrophic failure
- Fungible computation across arbitrary compute nodes
- Supports function as a service development patterns through computation outsourcing across distributed network

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY



Impact

ADCD offers a number of practical benefits on a global scale.

- Data-centric protections end the era of the "catastrophic" breach
  - ➢ Compromising a server does not degrade the confidentiality or integrity of the data stored on it
- Fungibility of secure execution allows users to know exactly how their data is being treated regardless of where the computation occurs
- Confidentiality policies enable users to allow others to manipulate their data without any information leakage
  - ➢ Fine-tuned policies can specify everything from what functions can be run on data, to who can retrieve output at any level of detail
- Universal provenance storage allows for both thorough forensic evaluation and real-time integrity policy verification

Next Steps

Currently, ACDC's design is being finalized. It has a Rust-based prototype implementation that supports confidentiality policies. Additionally, there is a framework for data provenance using an implementation of the Hyperledger-Fabric blockchain.

Our next steps include:

- Integrating functional execution using secure computation enclaves
- Creating and evaluating meaningful integrity/provenance policies for critical power systems
- Creating applications in our Rust framework for realistic, performant simulation and demonstration
- Integrating support for computation nodes using Multi Party Computation services

Acknowledgments:, Parker Diamond, Gene Itkis, Tyler Kaczmarek, Roger Khazan, Nabil Schear, Emily Shen, and David Stott