CIRI | CRITICAL INFRASTRUCTURE RESILIENCE INSTITUTE

A DEPARTMENT OF HOMELAND SECURITY CENTER OF EXCELLENCE

Advisor: Dr. Dan Tamir

USCG Mentor: Blair Sweigart

Kian O'Ryan

Maria Tomasso

Blockchain: Harnessing Linked Distributed Ledger

DHS Challenge

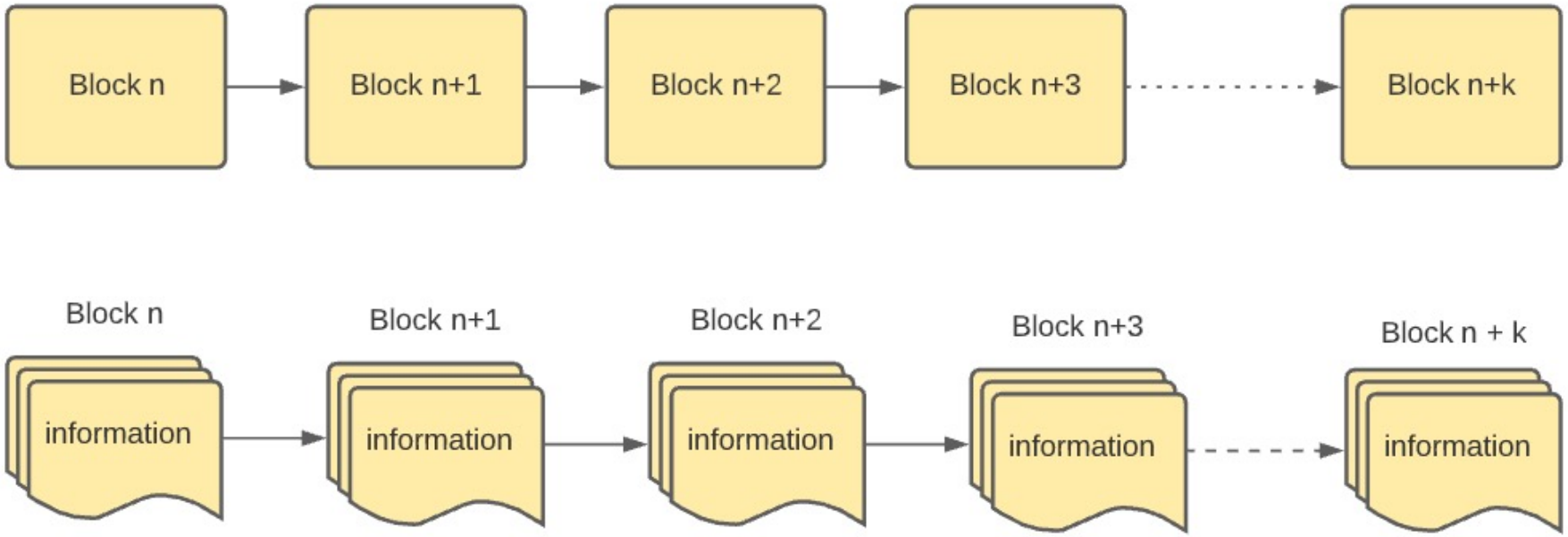
- Distributed ledger technologies (DLT) have received widespread attention in the past decade for their role in cryptocurrencies,
- Recently, they have shown promise in enterprise applications such as supply chain management, finance, and digital ownership.
- Many organizations are investigating the usefulness of DLT in their domain and parsing its true capabilities from the hype, including institutions within the United States government.
- This summer, we worked with the United States Coast Guard (USCG) to study important aspects of DLT and prepare a summary for use by USCG leadership. We have:
  - Studied the foundations of DLT
  - Analyzed DLT platforms developed for enterprise
  - Examined deployed DLT applications in USCG areas of interest
  - Developed novel applications of smart contracts for USCG missions
  - Researched compression and encryption in tandem as a potential security enhancement to the blockchain

DLT Background

DLT Background

- DLT is a **decentralized** system of accounting in which every participant maintains a full copy of the ledger, and all ledgers are kept in synch via **consensus mechanisms**.
- **Blocks** store sets of transactions. The blocks are joined by cryptographic links, which prevent tampering with past transactions. The entire ledger is made up of these linked blocks, hence the term '**blockchain**'.
- An **asset** is any currency or object managed by a DLT.
- The early DLTs used for cryptocurrencies (Blockchain 1.0/2.0) were **permissionless,** meaning anyone could join the system and contribute to consensus.
- In enterprise applications (Blockchain 3.0), **permissioned** blockchains are preferred to maintain a higher level of trust. Only approved participants can join and interact with the chain. This mitigates some of the drawbacks of Blockchain 1.0 and 2.0 such as vulnerability to attacks and high electricity usage

DLT: Blockchain



Smart Contracts

- **Smart contracts** were introduced in Blockchain 2.0 and provide a mechanism to automate transactions fairly via DLT.
- They are self-executing units of code that can govern assets.
- Early uses include agentless escrow, while Blockchain 3.0 has further refined them to automate many types of processes

DLT for Enterprise with Use Cases for USCG

Hyperledger Fabric

- We have determined that Hyperledger Fabric (HLF) is the preferred platform for enterprise DLT tools at this time.
  - In (Nanayakkara, 2021) Hyperledger Fabric ranked the highest among 24 DLT platforms for its scalability, interoperability with existing software, ease-of-use, security, and community availability.
  - In (Vadgama, N., & Tasca, 2021), HLF was found to be both the most popular platform in enterprise and the platform with the highest percentage of market-ready projects.

Use Case 1: Seafood Traceability

- Illegal, unregulated, and unreported (IUU) fishing threatens sustainability, food security, and the livelihoods of coastal workers.
- Furthermore, an estimated 24.9 million people are being coerced to work in IUU fishing globally.
- Prior solutions include electronic monitoring of vessels within fisheries and laboratory testing on commercial seafood products to determine providence, but IUU fishing persists.
- DLT has been proposed as a method of increasing visibility within the supply chain and validating the providence of legitimate seafood products.
- By removing the financial incentives for IUU fishing, this system would also address the crimes associated with IUU fishing such as forced labor.

Use Case 1: Seafood Traceability

- The IBM-developed **Food Trust** is HLF-based DLT designed to enhance traceability in food supply chains.
- Pilot studies are currently being evaluated by Walmart, Nestle, and Unilever with promising initial results.

Use Case 2: Inventory Management Systems

- Aircraft spare parts management (ASPM) requires extensive record keeping
- ASPM also involves tracking between entities to ensure safety and efficiency of operations.
- Current ASPM practices are labor-intensive with a risk of errors
- DLT could reduce errors and enhance data integrity between organizations.

Use Case 2: Inventory Management Systems

- Honeywell has implemented a blockchain-based market called **GoDirect Trade** for aircraft spare parts.
- Honeywell credits the scalability of the online market to automation via smart contracts.

Honeywell
GoDirect Trade

Use Case 3: Counterfeit Detection

- An estimated 10% of drugs in low- and middle-income countries and 1-2% of drugs in high-income countries are counterfeit, posing serious and sometimes lethal risks to patients.
- The issue is compounded by a fragmented supply chain making recalls more difficult once a fraudulent drug has been identified.

Use Case 3: Counterfeit Detection

- **Mediledger** is an HLF based system that allows patients and health officials to track drugs throughout the supply chain to ensure authenticity and aid with recalls if a counterfeit or contaminated products are identified.
- Gilead, Bayer, and Pfizer are currently involved with pilot testing for this system.

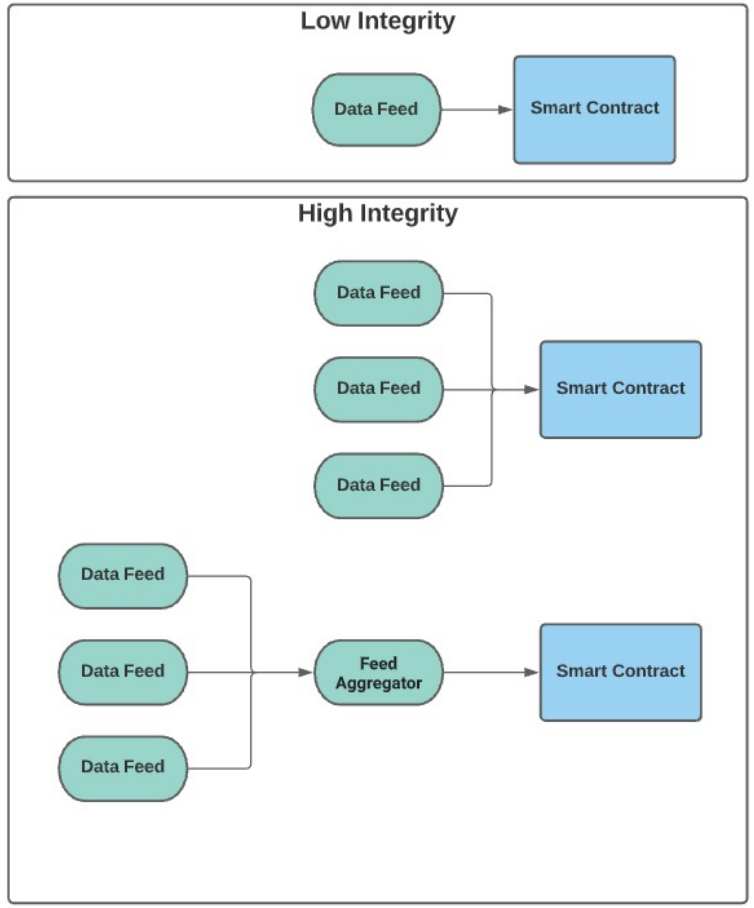
Conclusions

- Permissioned blockchains have utility towards USCG missions
- The immutable nature of DLT provides a strong foundation for enhancing traceability in supply chains
- Smart contracts can automate processes that would otherwise be time-consuming and prone to errors
- Fully developed platforms exist for implementing enterprise DLTs, many of which are interoperable with existing systems

Harnessing Smart Contracts

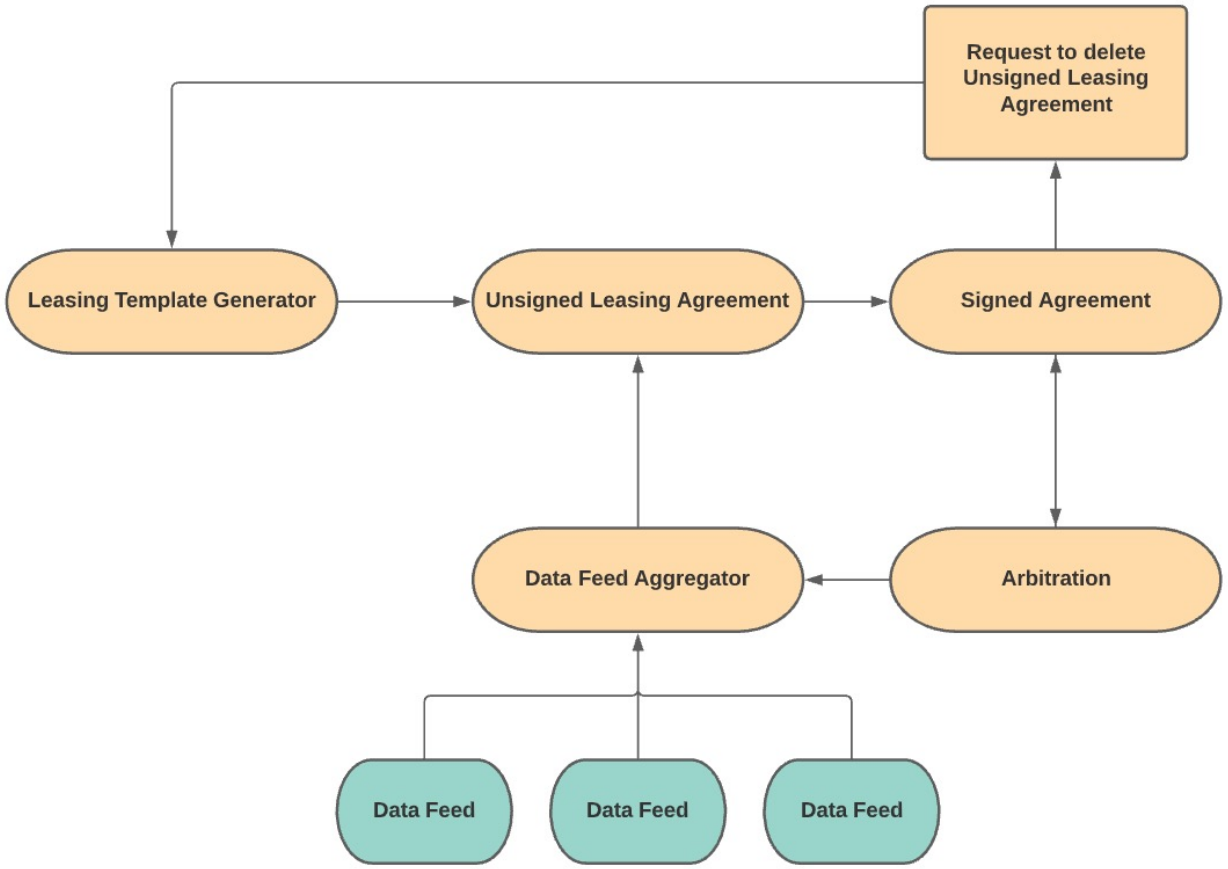
Blockchain : Smart Contracts

- The datum within each block may contain code for executable programs.
- The code for executable programs is mapped to specific accounts. These accounts are called Smart contracts.
- Smart contract (SC) accounts may receive external to blockchain information via data feeds.
- Data feeds information integrity is not guaranteed; data feed aggregation can help improve information accuracy.



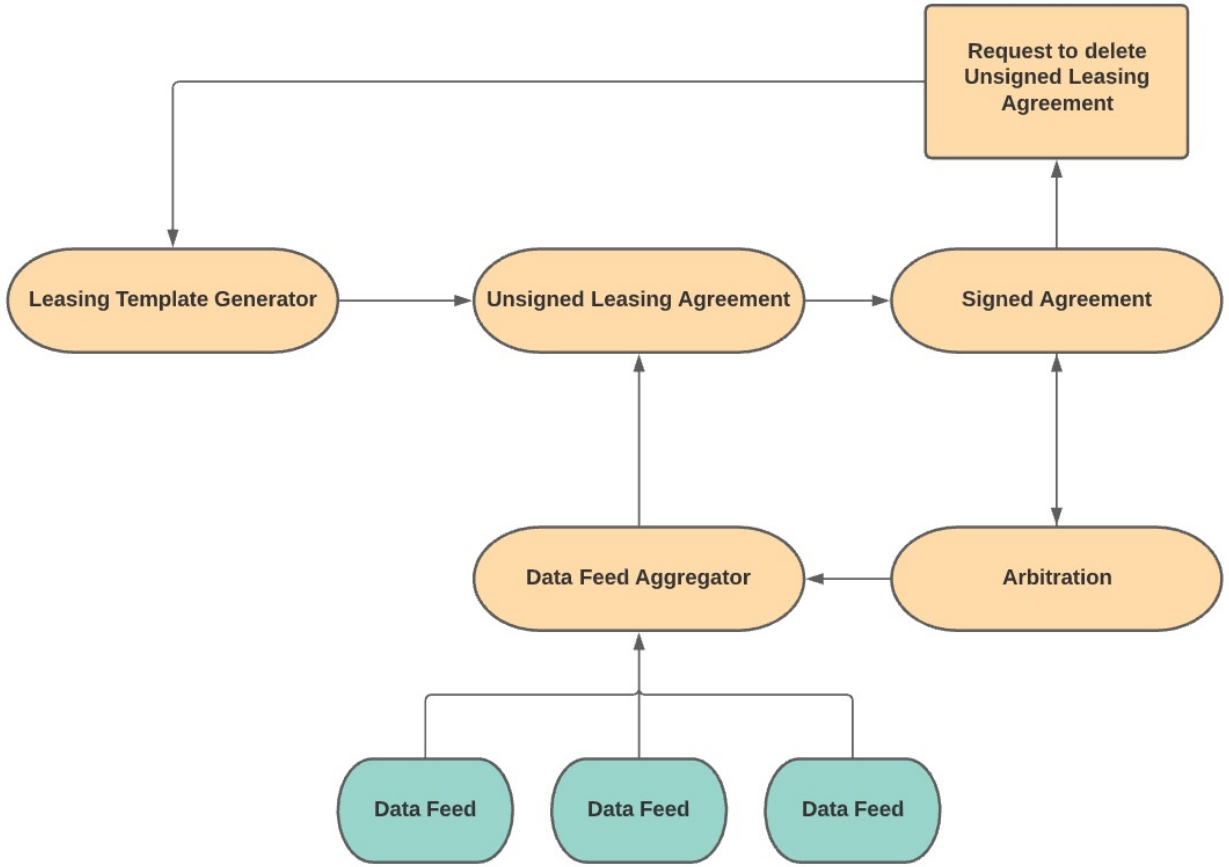
Leasing Agreement Smart Contracts via DLT

- Signing parties contact the Leasing Template Generator
- Leasing Template Generator receives lease terms from the leaser and lessee and generates an Unsigned Leasing Agreement (ULA)
- Data Feeds Aggregator relies on external feeds to map accounts to real identities.
- Once signing parties are authenticated ULA generates a signed agreement.
- The signed agreement has no contract owner and can not be deleted.



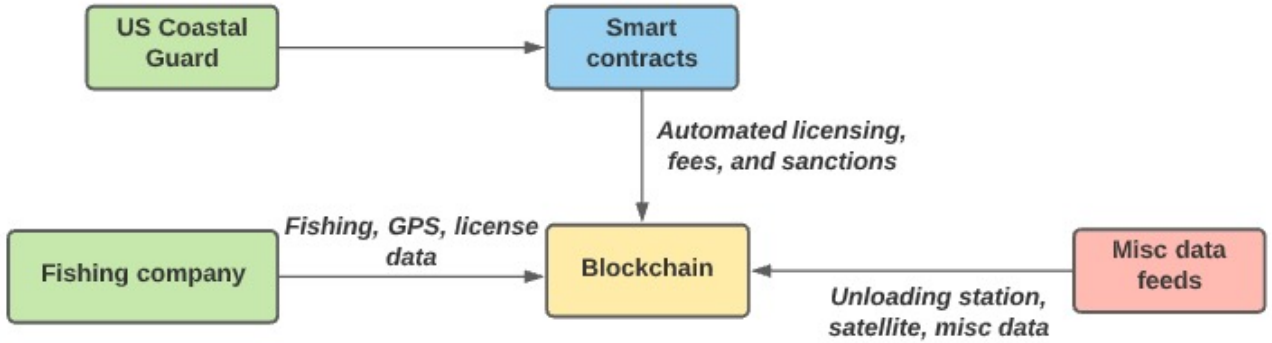
Leasing Agreement Smart Contracts via DLT

- Once signing parties are authenticated, ULA generates a signed agreement.
- To liberate network resources, the signed agreement SM asks the leasing template generator to delete the unsigned leasing agreement.
- The signed agreement has a built-in arbitration function that signing parties may use to request arbitration to resolve disputes.
- The data feed aggregator is provided with arbitration results.



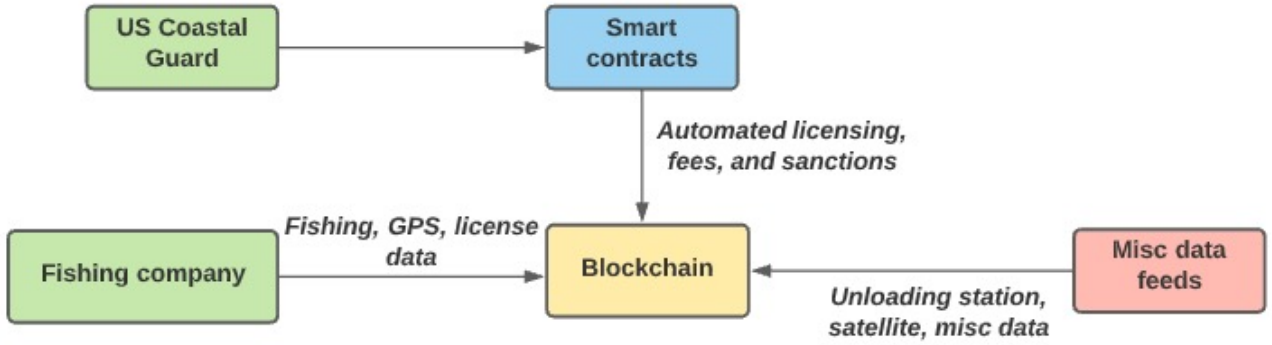
Fishing Smart Contracts via DLT

- Fishing companies are required to provide specific data to the blockchain. This includes fishing data, GPS coordinates and licenses.
- USCG creates and controls smart contracts that monitor the blockchain.
- Fishing companies provide with the blockchain with their documentation, once verified by a smart contract, a fishing license can be automatically issued.



Fishing Smart Contracts via DLT

- Smart contracts can monitor GPS and fishing data and issue warnings, fines and sanctions based on any detected discrepancies.



Improving Shannon-Fano-Elias Rate and Encryption Resilience

Introduction

- The traditional method for preforming compression and encryption of data is to first compress the data with a given data compression method, then encrypt the compressed data.
- Tandem data compression and encryption might be more efficient with respect to throughput than performing compression and encryption separately.
- Shannon Fano Elias (SFE) code is a leading candidate for this method, but its compression capabilities are not as competitive as other compression techniques such as Huffman coding and Lempel Ziv-based coding.

Terminology

- FLC, VLC, and uniquely decodable code
- The **average source code length (code rate) is given by:**

$$L(s) = \sum_i p(a_i) \times l(a_i)$$

- **It is measured as bits/symbol.**
- **The entropy of a source is given by:**

$$E(s) = \sum_i p(a_i) \times log_2((P(a_i))$$

- **According to Shannon theory, the entropy is the lower bound on code rate.**

Permutations on symbol-order

- Huffman code enables a limited number of permutations of the order of symbols (hence their code-words) that yield the same rate.
- Theoretically, for SFE compression, each permutation on symbol order can yield a set of different code words with same rate.
  - Hence with *n* symbols there might be as many as *n!* permutations
  - Our research is the first to demonstrate that the number of effective permutations is much lower than the upper bound of *n!*.
- Hence, SFE can be used in private key encryption where the permutation is the key.
  - The resilience of the schema is a function of the number of permutations.
- In general, however, SFE coding rate is inferior to Huffman coding rate.

Symbol	Probability	Huffman-A	Huffman-B	SFE-A	SFE-B
B	0.25	10	10	100	100
A	0.125	110	111	101	110
D	0.5	0	0	00	01
C	0.125	111	110	110	110

Rate of Entropy Codes

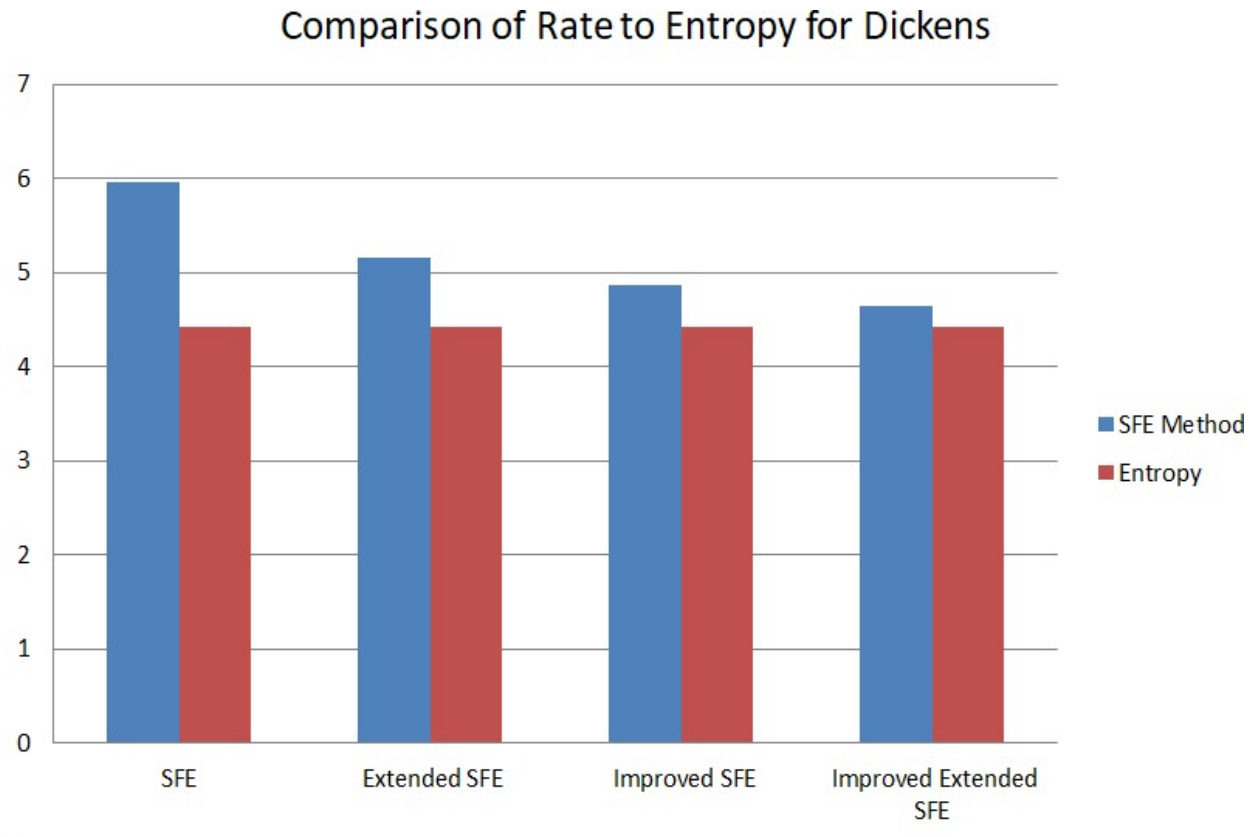
- The **improved SFE** uses an algorithm, developed by Tamir, to reduce the average length of SFE coding.
- The **extended Huffman / SFE** uses an extended alphabet by producing all the combinations of *k* symbols and applying Huffman / SFE to the new alphabet.
    - Hence the number of symbol order permutations increase exponentially with the number symbol combinations.
- The **improved extended SFE** runs the improved SFE algorithm on an extended SFE alphabet.
- Huffman code is bounded by Entropy + one bit. SFE is bounded by entropy + two bits. The extended versions can approach entropy.
- Often, the improved SFE algorithm eliminates one bit from each code word while maintaining UD.

Experiments

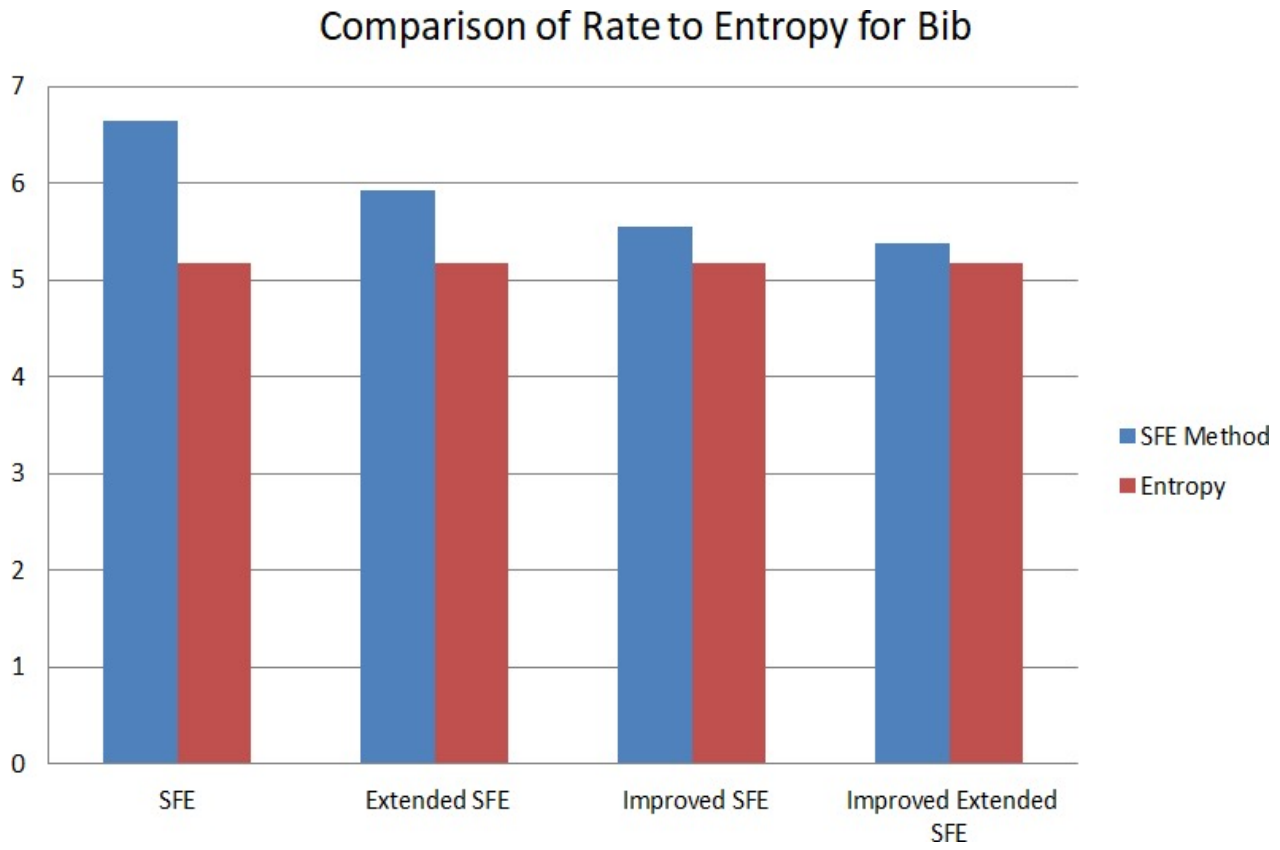
We did the following steps to test each method of SFE:

1. Implemented each variant.
2. Ran each variant on all the files from the Silesia and Calgary Benchmarks.
3. Compared their average lengths and effective number of symbol order permutations (resilience).

Results


Comparison of Rate to Entropy for Dickens
7
6
5
4
3
2
1
0
SFE
Extended SFE
Improved SFE
Improved Extended SFE
SFE Method
Entropy

Results (cont.)



Extended SFE with three Combinations

- The alphabet becomes too large.
- Probabilities become too small.
- Some combinations may have the same code word.

Future work

- Explore the dynamic SFE and arithmetic coding.

Acknowledgement

This research was performed under an appointment to the U.S. Department of Homeland Security (DHS) Science & Technology (S&T) Directorate Office of University Programs Summer Research Team Program for Minority Serving Institutions, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS. ORISE is managed by ORAU under DOE contract number DE-SC0014664. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of DHS, DOE or ORAU/ORISE.