LibFTE: A Toolkit for Constructing Practical Format-Abiding Encryption Schemes

Authors: Daniel Luchaup, Kevin Dyer, Somesh Jha, Thomas Ristenpart, Thomas Shrimpton

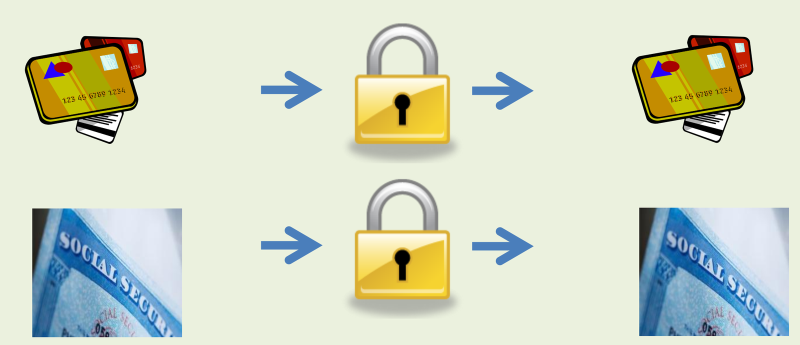
Formatted Encryption

- Traditional Encryption: encrypts formatted data as an unformatted sequence of bytes
- Sometimes the encrypted data must be formatted (for instance legacy applications)
- Formatted Encryption: encrypts formatted data as formatted ciphertext
- ...but Formatted Encryption only works for regular languages, when DFA fits memory. Awkward to use.

Format Preserving Encryption

(Bellare et. al., 2009)

FPE: plain text and cipher text have similar format

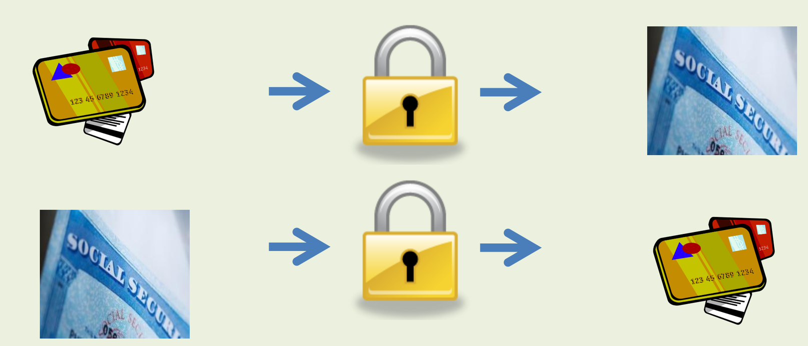


Applications: legacy databases, payment industry

Format Transforming Encryption

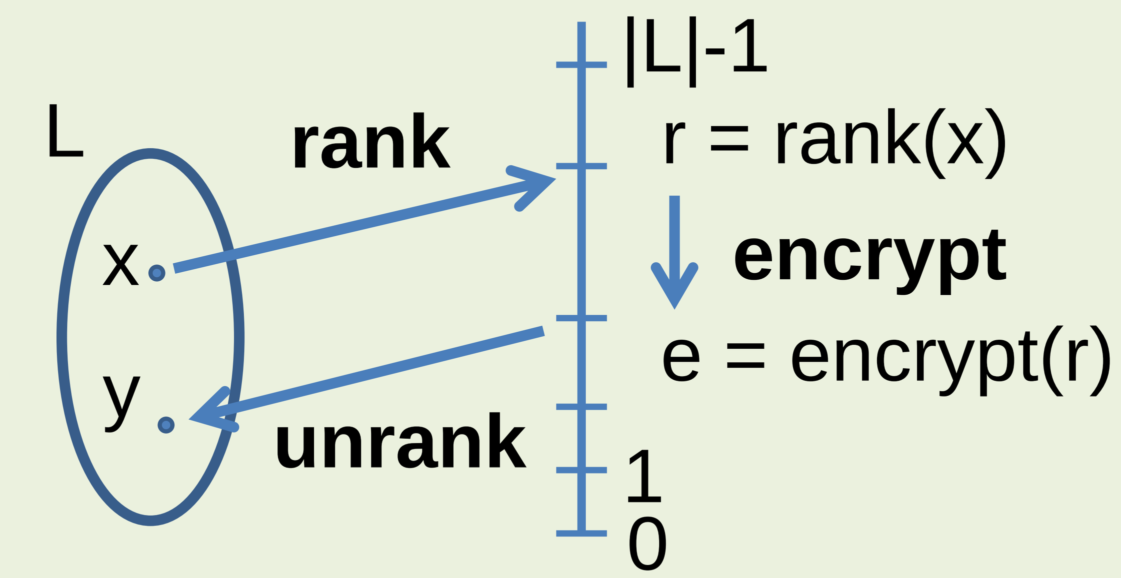
(Dyer et. al., 2013)

FTE: plain text and cipher text have different format



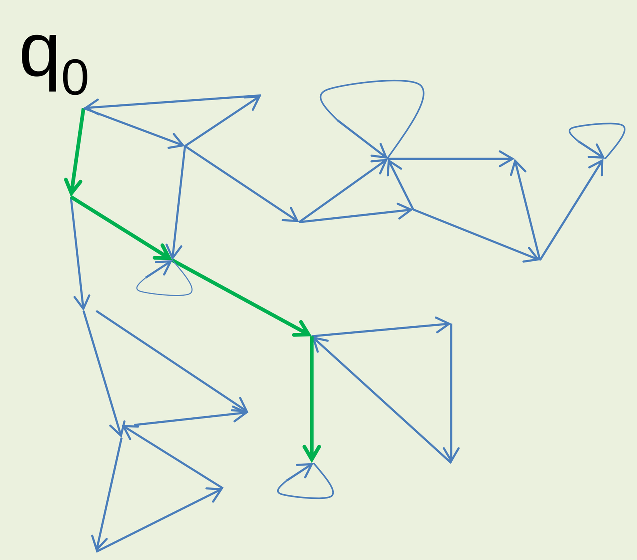
Applications: censorship avoidance
Example: Tor encrypted as HTTP

Rank-And-Encipher FPE



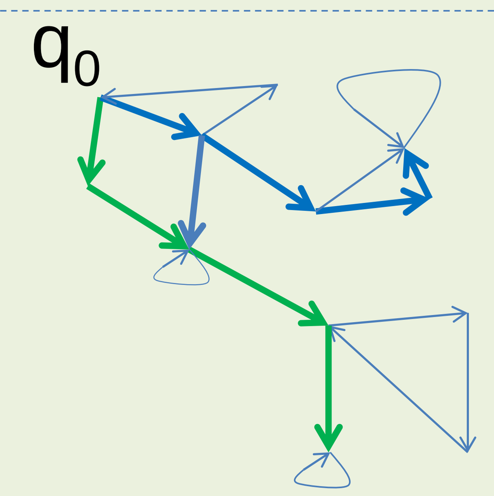
Rank bijection: $L \rightarrow Z_{|L|}$, unrank the inverse
FPE(x) = unrank(encrypt(rank(x)))

Ranking Regular Languages



Old: DFA based
- Count accepting paths
- Unique accepting paths

Issues:
- State space explosion



New: NFA based
- Count accepting paths
- Fewer states (works when DFA doesn't)

Issues:
- Possibly multiple accepting paths

Must account for the possibility of multiple accepting paths: use relaxed ranking

Relaxed Ranking

Rank: $L \rightarrow Z_N$, is injective
Unrank: $Z_N \rightarrow L$, is surjective
Unrank(Rank(x)) = x

Condition for correct decryption:
Rank(Unrank(r)) = r
Only holds for $r \in$ Rank(L)
Must adjust the rank-and-encipher

Use Cycle-Walking:
Repeat: r = encrypt(r)
Until Rank(Unrank(r)) = r

LibFTE

Public implementation
Generic framework, simple specification
- regular expression, size ranges

Fast
- Improved DFA ranking
- NFA ranking / Relaxed Ranking under the hood
- Choice of DFA/NFA ranking transparent to user

Configuration
- Input/Output language selection
- Tool to help user reasoning about configuration choices

Performance analysis
Applications:
- In browser encryption
- DB encryption and compression