Reconciling Two Views of Cryptography

(The Computational Soundness of Formal Encryption)

Martin Abadi, Phillip Rogaway

Presented by:
Bushra, Justin, Nydia, Ryon
Outline

- Goal of the Paper
- Formal View of Encryption Schemes
- Computational View of Encryption Schemes
- Bridging the Gap
Formal View

\{ M \}_K
Goals:

- How to convert between the formal view and the computational view.
- Proving equivalence implies indistinguishability.
Expressions

EXP

\( M, N ::= \) expressions

\( K \) key (for \( K \in \text{Keys} \))

\( i \) bit (for \( i \in \text{Bool} \))

\( (M, N) \) pair

\( \{M\}_K \) encryption (for \( K \in \text{Keys} \))
(\{(\{(0,K')\}_K\}_K',K)\)
Cyclic Expression

\[ (\{K'\}_K, \{K\}_{K'} ) \]
Acyclic Expression

Formal View

\((\{K\}_{K'}, \{0\}_K)\)
Equivalence

Two expressions are said to be equivalent if they have the same patterns:

\[ M \equiv N \text{ if and only if } \text{pattern}(M) = \text{pattern}(N) \]
Pattern

Equivalent to $\text{Exp}$, but with the inclusion of the □ symbol.

□ - represents a ciphertext that an attacker cannot decrypt.
Formal View

\[(\{\{K_1\}\}_{k_2}^{k_3}, K_3) \equiv (\{0\}_{k_1}^{k_3}, K_3)\]
Equivalent up to Renaming

Two expressions are said to be equivalent up to renaming if there exists a bijection, \( \sigma \), on \textbf{Keys}.

\[ M \sim N \text{ if and only if } M \equiv N\sigma \]
Formal View

\[(\{0\}_K, K) \cong (\{0\}_{K'}, K')\]
Recap

Formal View

Expression
- Acyclic
- Cyclic

Patterns
- Equivalent
- Equivalent up to renaming
Computational View

Encryption Scheme: $\Pi (K, \epsilon, D)$

- $K$: Parameter $\times$ Coins $\rightarrow$ Key
- $\epsilon$: Key $\times$ String $\times$ Coins $\rightarrow$ Ciphertext
- $D$: Key $\times$ String $\rightarrow$ Plaintext

where Parameter is represented by $\eta$
Computational View

**Ensemble**

A collection of distributions on strings

$$D = \{D_n\}$$
If an adversary cannot tell what set $x$ is from then $D$ and $D'$ are indistinguishable.
Attributes of Secure Encryption Schemes

- Repetition Concealing/ Revealing
- Which-key Concealing/ Revealing
- Message-length Concealing/ Revealing
Repetition Concealing/Revealing

- Given $c = E_k(x)$ and $d = E_k(x)$
- Can you tell whether the plaintext $x$ is the same in both instances?
Computational View

Which Key Concealing/Revealing

- Given $c = E_k(x)$ and $d = E_{k'}(x)$
- Can you tell whether $k$ and $k'$ are the same?
Computational View

Message Length
Concealing/Revealing

- Given $c = E_k(x)$ and $d = E_k(y)$
- Can you tell whether $x$ and $y$ are the same length?
Security Types

Represented as a 3-bit binary number, where concealing = 0 bit and revealing = 1 bit

Example:

- type-0 : 000  Repetition, Which-key and Message-length Concealing
- type-3 : 011  Repetition concealing, Which-Key and Message-length Revealing
Computational View

Type-n Advantage

Oracle

Good Box

Bad Box

$\epsilon_K(m)$

Oracle response

$\epsilon_K(\text{garbage})$

$m$
Computational View

**type-0 Security Advantage**

![Diagram showing the computational view of type-0 security advantage](image-url)
type-1 Security Advantage

Computational View

Oracle

$\epsilon_K(m), \epsilon_{K'}(m)$

$\epsilon_K(0^{\|m\|}), \epsilon_K(0^{\|m\|})$

Oracle response

$\epsilon_K(m)$ or $\epsilon_{K'}(m)$

$\epsilon_K(0^{\|m\|})$ or $\epsilon_K(0^{\|m\|})$
Computational View

type-3 Security Advantage

Oracle

\[ \epsilon_K(m) \]

\[ \epsilon_K(0^{\mid m \mid}) \]

\( m \)

Oracle response

\( \epsilon_K(m) \)

\( \epsilon_K(0^{\mid m \mid}) \)
Computational View

CTR mode is which–key concealing, message length revealing, repetition concealing

- Which-key, repetition
  - Cannot tell psuedorandom function from a random function
- Ciphertext length is same as plaintext
Hiding Message length for CTR?

- Make the plaintext some fixed length
- Then the plaintext is encrypted
Recap

Computational View

Encryption Scheme: $\Pi (K, \epsilon, D)$

Repetition

Which-Key

Concealing

Revealing

Message-length

Advantage
Bridging the Gap

Relating the two views of Cryptography

Step 1. Associate an ensemble to an expression $M$, given an encryption scheme $\Pi$.

Step 2. Proving equivalent expression implies indistinguishable ensembles
Given:
formal expression $M \in \text{EXP}$
encryption scheme $\Pi (K, E, D)$

Then:
$[M]_{\Pi[\eta]}$ • Distribution of strings
$[M]_{\Pi}$ • Ensemble
Formal expression

- **KEY**
  - mapped to bits
  - Each expression in the pair is separated and fed in to the algorithm individually

- **Pair (X,Y)**

- **{M}_K**
  - M is fed back in
  - Mapped to corresponding bits used to represent True or False

- **Bool**
Bridging the Gap

Formal expression: \( \{\text{true}\}_K, K \)

\[
\begin{align*}
\text{KEY} & \quad \rightarrow \\
\{\text{true}\}_K & \\
\rightarrow & \\
\text{Pair} & \quad (\{\text{true}\}_K, K) \\
\rightarrow & \\
\{M\}_K & \\
\rightarrow & \\
\{\text{true}\}_K & \\
\rightarrow & \\
\text{Bool} & \\
\rightarrow & \\
1 & \\
\rightarrow & \\
\text{true} & \\
\rightarrow & \\
(\{\text{true}\}_K, K)& 
\end{align*}
\]
“Let $M$ and $N$ be acyclic expressions and let $\Pi$ be a type-0 secure encryption scheme. Suppose that $M \simeq N$. Then

\[
\begin{bmatrix} M \end{bmatrix}_\Pi \simeq \begin{bmatrix} N \end{bmatrix}_\Pi
\]

Proof to come….stay tuned…..