Reading Guide 11: Reconciling Two Views of Cryptography

Justin, Nydia, Ryon, Bushra

November 19, 2004

This paper discusses two abstract views of cryptography: formal and computational. The formal view uses functions and expressions with atomic primitives that are not realizable on a digital computer. The computational view treats the key, plaintext, and ciphertext as strings of bits. In this model the security properties of a cryptographic protocol are based on probability and computational complexity. The paper presents a theorem that bridges the gap between formal and computational cryptographic models.

Section 2: The formal view uses notation for cryptographic expressions in such a way that M is the plaintext message and K is the key forming$\{M\}_K$ as a formal expression. While this has the advantage of being a simple model, a proof does not guarantee security. Next, an overview of the computational model is presented. A computational encryption scheme is defined as $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where $\mathcal{K}$ is the key generator, $\mathcal{E}$ is the encryption algorithm, and $\mathcal{D}$ is the decryption algorithm. The random coins r are utilized to make the encryption scheme probabilistic rather than deterministic. The security of the computational view is based on the advantage of an adversary. If the maximal advantage is minimized than the encryption scheme is regarded as ”good”.

Section 3: 3.1 presents a grammar for use in formal encryption that is used later in the paper. Formal expressions deal with atomic symbols. Additionally, the paper defines a formal expression as cyclic or acyclic — cycles are generally not a good cryptographic practice. 3.2 presents entailment, patterns, and 2 differing forms of equivalence. $\vdash$ is the entailment operator where $M \vdash N$ is read M entails N. This means that N can be determined from M. Patterns **Pat** is an extension of **Exp** that includes $\Box$, which represents a ciphertext that an attacker cannot decrypt. The relation equivalence up to renaming is denoted $\cong$. This is less strict than logical equivalence denoted $\equiv$. 3.3 provides some examples of those items defined in this section.

Type	Bits	Repetition	Which-Key	Message-length
type-0	000	concealing	concealing	concealing
type-1	001	concealing	concealing	revealing
type-3	011	concealing	revealing	revealing

Table 1: Table representing three different types of encryption scheme security.

Section 4: In 4.1, note that × is the cartesian product read as cross. 4.2 presents different attributes of encryption schemes. These are summarized in the table 1. In 4.3, the security parameter η (eta) determines the size of the key. This section presents advantages of three types of encryption. Each advantage can be thought of as the probability that an adversary can distinguish between a box that uses the adversary's query and a box that ignores the adversary's query. 4.4 points out that CBC and CTR encryption are which key concealing. Also, to make a which key and message length concealing encryption scheme, simply encode all plaintext into a fixed length before encryption.

Section 5: This section includes the main purpose of the paper: mapping the formal model to the computational. In 5.1 every formal expression is mapped to a distribution of strings, $[M]_\Pi$. The algorithm begins by initializing every key in the formal expression M to a bit string. The convert algorithm parses the formal expression to its corresponding bit strings. The theorem essentially states that an expression that is equivalent up to renaming implies that their corresponding bit string ensembles are indistinguishable. **You do not need to read the proof of the theorem.**