CS 512 Lecture Notes

Yara Awad

April 6, 2018

***EasyCrypt*: Recap**

EasyCrypt is a framework specialized for generating proofs of security protocols. A proof is structured as a sequence of one or more lemmas, or goals. A goal is proven using tactics. Tactics are logical rules that represent general reasoning principles. Each tactic transforms a goal into zero or more subgoals. When no further subgoals can be generated, the lemma, or goal, has been proven.

EasyCrypt runs proofs inside a proof engine. A proof engine recognizes goal lists. Every goal in a goal list consists of a *context* and a *conclusion*. The context of a goal consists of type variables and a set of assumptions related to that goal. The conclusion of a goal is a single formula that we wish to prove to be true.

Security of a cryptographic protocol can be modeled using *games*. This security can be proven using a *sequence of games* approach. EasyCrypt models cryptographic games as modules. A module consists of global variables and procedures.

***EasyCrypt*: Symmetric Encryption from Pseudorandom Functions**

Encryption Scheme Correctness and Security

Encryption protocols are used to transfer data/messages securely between parties. The symmetric encryption protocol consists of an encryption scheme in which the same key is used to encrypt a plaintext and to decrypt the corresponding ciphertext. An encryption scheme can be correct and/or secure:

- An encryption scheme is correct if, for some plaintext m and for some key k, the encryption of m using k produces ciphertext c, where then decryption of c using k always produces m.
- An encryption scheme is secure if no adversary is able to break the scheme with non-negligible probability.

The security of an encryption scheme may be proven within EasyCrypt using the *sequences of games* approach. In this approach, and adversary A has access to the encryption scheme. That is to say, the adversary A has access to the encryption and decryption procedures and is allowed to call them some finite amount of times. If, in this game, encryption is probabilistic, then it is possible to prove bounds on the security of the encryption scheme.

Encryption Scheme Using Pseudorandom Functions

True random functions (TRFs) exhibit true randomness. A TRF $F : \ text \rightarrow text$ is a map that takes as input some text variable and returns as output some text variable. The relationship between the input text and the output text is truly random. F builds a finite map of $(text \rightarrow text)$ mappings, where for every new input text, F generates a new random output text and adds this mapping to its map. In some sense, a TRF implements *memoization.*

Pseudorandom function (PRFs) exhibit pseudo-randomness. That is, no efficient adversary A can distinguish between a PRF and a TRF with non-negligible advantage. A PRF $F_{pseudo} : text \rightarrow text$ is a deterministic function that takes as input some text variable and returns as output some text variable. That is, for a given input text x_{in}, F_{pseudo} always returns x_{out}.

The space of TRFs can be calculated to be of size $|text|^{|text|}$, where $|text|$ is the size of the space of all text variables. The space of PRFs can be calculated to be of size $|key|$, where $|key|$ is the size of the space of all possible keys generated by a PRF. The above makes secure PRFs a good tool for cryptographic encryption schemes.