Proofs in Cryptography*

Ananth Raghunathan†

Abstract

We give a brief overview of proofs in cryptography at a beginners level. We briefly cover a general way to look at proofs in cryptography and briefly compare the requirements to traditional reductions in computer science. We then look at two security paradigms, indistinguishability and simulation based security. We also describe the security models for Secret Key and Public Key systems with appropriate motivations. Finally, we cover some advanced topics and conclude with a few exercises and hints.

1 Introduction

TALK ABOUT THE NEED FOR WORKING WITH REDUCTIONS AND DIFFERENT SECURITY DEFINITIONS. THE SUBTLETIES THAT ARISE IN DEFINITIONS, THE TIME IT TOOK TO COME UP WITH GOOD DEFINITIONS OF SECURITY.

Before we go about describing a general outline to prove the security of constructions, let us have an informal discussion. Proving the impossibility of a particular computational task is extremely difficult. It is very close to the P vs. NP question (add reference) that is the central question in computer science today. It becomes doubly difficult when do not know what class of computational tasks we are allowed to utilize (unlike the case of P vs. NP, where a Turing machine has been rigorously and beautifully formalized). Therein lies the central difficulty in proving the security of cryptographic constructions.

One way to address the latter question is to explicitly model what an adversary who intends to break our cryptosystem is allowed to do. Notice this requires us to define at least two things: what do we mean by an "adversary" and what it means to "break" the cryptosystem. A lot of emperical and theoretical work goes into

*__Rough Draft__ of a handout for CS255: Introduction to Cryptography by Dan Boneh.

†ananthr@stanford.edu

the current models of "adversary" and "break" and currently the definitions we use have been used widely in practice and give us good results emperically.

However, as it happens often in this field, the adversary in the real world is not constrained to behave as we dictate him[1]. There have been several real world attacks (side-channel attacks, cold-boot attacks are recent innovative attack (add references)) that completely side-step the adversarial model and as expected completely breaks the security of the scheme. However, this only serves as motivation to model stronger adversaries and thereby construct more robust cryptosystems that are provably robust (with respect to the new models).

In conclusion, remember two important things: To analyse a cryptosystem you must define an adversary model and a security game.

1. **Adversary model:** This defines formally the power of the adversary. It includes specifics whether the adversary is deterministic/randomized, uniform/non-uniform, interactive/non-interactive and how he interacts with the security game.

2. **Security game:** This defines formally the power the adversary has over the cryptosystem. Whether he has access to a single ciphertext, multiple ciphertexts, multiple keys, etc. It also defines when an adversary is said to break the system.

Often adversaries are modeled similarly, but depending on whether you want weak or strong security guarantees, you modify the security game accordingly.

Two broad paradigms for security games are indistinguishablity games and simulation games. Subsection 1.3 will talk about it in more detail.

[1] or her. I will restrict myself to male pronouns for simplicity.

5 Exercises

1. Prove that truncating the output of a PRF is still secure
2. Prove that any subset of the bits of a PRG are still indistinguishable

3. Prove that a ℓ-expanding PRG is a small domain PRF
4. Introduce hardness of SVP in perp lattices and prove that $f_A(x) = A \cdot x \pmod{q}$ is a collision resistant hash function
5. Prove that a PRF implies a One Way Function