Problem 1: Secure DNS

DNS is a major point of vulnerability in the Internet architecture. In particular, a man-in-the-middle attacker can inject false DNS name bindings by “hijacking” DNS requests. This question asks you to propose a scheme to prevent such an attack by distributing asymmetric key pairs along the DNS hierarchy of trust. Your scheme should allow a client (or equivalently a local DNS server acting as a resolver proxy) to verify the authenticity and integrity of each DNS translation, i.e., to validate efficiently that a name binding was established by a chain of DNS servers authorized to resolve that name.

a. What structures does your scheme add in each DNS response? How does the sender generate this new information, and what does the receiver do with it? Include any other details necessary to understand your scheme. (35 points)

b. How does your scheme apply to both iterative and recursive lookups? (5 points)

c. What is the impact if an attacker successfully compromises a DNS server, i.e., how much damage can the attacker do? (10 points)

d. What is the impact if an attacker successfully compromises a DNS resolver proxy, i.e., how much damage can the attacker do? (10 points)

Problem 2: Quantitative Web Caching

Sketch the following “back of napkin” graphs, guided by the Web cache modeling principles (e.g., Zipf distributions) given in the Rabinovich/Spatscheck text and the paper On the scale and performance of cooperative Web proxy caching [Wolman, Voelker, et al, SOSP 1999]. Be sure to label each axis, e.g., indicate whether each axis is on a log or linear scale, and the maximum/minimum values. Note that it is not necessary to recall the formulas to draw these graphs.

a. Probability of reference to a given Web object as a function of its popularity rank. Draw three curves on the graph for different values of the Zipf $\alpha$ popularity parameter. (15 points)

b. Probability of reference to a given Web object as a function of the number of objects in the “universe”. Draw at least three curves on the graph to show the effect of the object’s rank and the $\alpha$ popularity parameter. (10 points)
c. “Ideal” cache hit ratio as a function of population size. Draw enough curves on the graph to show the effect of the number of objects in the object universe, the $\alpha$ popularity parameter, and the per-client request rate. Add a bullet list of real-world factors that might affect this “ideal” prediction. (20 points)

d. “Ideal” cache hit ratio as a function of population size for two scenarios: (i) a demand-side caching scheme in which the client load is balanced arbitrarily across two isolated proxy caches, and (ii) a Content Distribution Network (CDN) scheme in which the object universe is distributed evenly across two proxy cache surrogates. (15 points)

Problem 3: Systems and Assumptions

Consider the following systems in which a group of nodes cooperate to provide a scalable service: Porcupine, DDS, CFS, CARP (hash-based Web caching), and LARD. All of these systems use some variant of hashing and/or directories to distribute work across the participating nodes. The designers of these systems made varying assumptions about the environments in which their systems function. For each assumption listed below, list the systems that depend on the assumption, then name at least one of the systems that does not depend on it, and summarize the properties that enable the system to operate without that assumption.

a. Network partitions do not occur. (15 points)

b. Node failures are always fail-stop; no byzantine failures may occur. (15 points)

c. Each node knows the identities of all participants. (15 points)

d. Some node knows the identities of all participants. (15 points)

Problem 4: Your Project

In a short paragraph, outline your proposed semester project for CPS 212. (20 points)