# 1 Hello, David

These are pretty much just my notes. I'm writing them in LaTeXmostly for practice. This can probably transform into a final document, so feel free to make changes on any level. I'm not even trying to word this very well.

# 2 Desirable Properties

There are three main properties that a proxy-anonymization system might want to satisfy.

## 2.1 Anonymity to Server

The server cannot determine the identity of the client.

## 2.2 Encrypted to Proxy

The proxy has no way of figuring out the data. This will of course require that the client get some sort of public key for the server.

## 2.3 Server Unawares

The server is unaware that it is being commuicated with via a proxy.

# 3 Scenarios

In this section, we present scenarios that would warrant the properties above.

## 3.1 Proxy

## 3.2

# 4 Single Proxy

In this section we present a single proxy.

1. Generate

# 5 PDRSA

## 5.1 Algorithm

Here we present the algorithm for Piecewise-Decrypt RSA. The main difference now is that there are now two private keys, one of which is kept and the other of which is passed to the proxy. I forgot why this distinction is important. I may have wasted a few hours on this.

### 5.1.1 Key generation

Let $p$, $q$ be large primes like in RSA. Define $n = pq$. Note that $\phi(n) = (p-1)(q-1)$. Now select $e$, $d_2$ that are coprime with $\phi(n)$. This implies that $ed_2$ is also coprime with $\phi(n)$. While $e$ can be selected to minimize the cost of encryption, $d_2$ must be selected randomly. Let $d_2 = (d_1 e)^{-1} \mod (p-1)(q-1)$.

Publish $(e, n)$ as the public key. $(d_2, n)$ is the semiprivate key. $(d_1, n)$ is the private key.

### 5.1.2 Encryption

$c = m^e \mod n$

### 5.1.3 First Pass Decryption

$h = c^{d_2} \mod n$

### 5.1.4 Final Decryption

$m = h^{d_1} \mod n$

## 5.2 Proof of Correctness

Pretty similar to the one on wikipedia for RSA. Fermat's is easier to understand, but Euler's is shorter.