

CONFIG.PHP:

```
#!/usr/local/bin/php -q

<?php
/*developed using php 4.3.2*/

include 'funcs.php';

error_reporting (E_ALL);

/*set the time limit to infinity, so we never stop waiting for a connection*/
set_time_limit(0);

/* turn on implicit output flushing so we see what we're getting as it comes in */
ob_implicit_flush();

$port_lf = 1108; //port for listening for connections, and forwarding on packet to $next_neighbor

$tmpdir = "/tmp/ring/";
$log = $tmpdir . "log";
$del_cmd = "rm -f " . $tmpdir . "*";
system($del_cmd);

$buf = 32000;

/*MySQL info*/
$mysql_db = "localradio";
$mysql_host = "localhost";
$mysql_delim = "::"; //mysql delimiter

$continue = TRUE;
if(file_exists("config.dat.pgp")) {
    print "A config file already exists. Would you like to continue? (y/n): ";
    $continue_yn = trim(fgets(STDIN,256));
    if($continue_yn == "y")
        $continue = TRUE;
    else
        $continue = FALSE;
}
```

```

if($continue) {
    for($i=0; $i<20; $i++) {print "\n";}
    print "Greetings! I assume this is the first time you have attempted\nto use this system."
        . "If this is not the case, then please make\nsure the file \"config.dat.pgp\" is in"
        . " the same folder as this script! \nThanks\n\n";
    print "Before continuing, please make sure you have completed the following:\n";
    print "\tYou know the name of the server that will \"introduce\" you \n\t\tinto the ring\n";
    print "\tPorts 1108 & 1208 are open on this machine\n";
    print "\tA MySQL database exists that is called \"localradio\" with a user that\n\" .
        "\thas privileges:\n\t\tselect, insert, delete, & update\n";
    print "\t\tteg, GRANT SELECT,INSERT,DELETE,UDPATE ON localradio.* \n\" .
        "\t\tTO <user>@localhost IDENTIFIED BY '<password>'\n";
    print "\tOn that same MySQL database, there's a user with just SELECT privileges:\n";
    print "\t\tteg, GRANT SELECT ON localradio.* TO <user>@localhost \" .
        "IDENTIFIED BY '<password>'\n";
    print "\tRun the mysql script to create tables (mysql_create_tables). \n\" .
        "\t(Must run as someone with create privileges)\n";
    print "\t\tuse: mysql --user=<username> -p <mysql_create_tables\n";
    print "\tA pair of PGP keys has been generated, and is located in:\n\t\t~/pgp/\n";
    print "Have all these requirements been met? (y/n): ";
    $ready = trim(fgets(STDIN,256));
    if($ready != "y") {
        print "\nokay, please rerun config.php when ready!\n";
        exit;
    }
}

do{
    print "Please enter the name of this server (no 'http'): ";
    $me = trim(fgets(STDIN,256));
} while( strlen($me) < 1 );
do{
    print "Please enter the name of the server that will introduce you to this ring"
        . "(once again, no 'http'): ";
    $intro = trim(fgets(STDIN,256));
} while( strlen($intro) < 1);

/*connecting to mysql*/
do {
    print "\nPlease enter the name to use to login to the local mysql db" .
        " (with full privs): ";
}

```

```

        $mysql_id = trim(fgets(STDIN,256));
        print "mysql pw: ";
        $mysql_pw = trim(fgets(STDIN,256));
    } while( ($mysql_link = mysql_connect($mysql_host, $mysql_id, $mysql_pw)) == FALSE);

/*selecting database*/
if( mysql_select_db($mysql_db) == FALSE) {
    print "could not select database: $mysql_db\n";
}
mysql_close($mysql_link);

/*connecting to mysql*/
do {
    print "\nPlease enter the name to use to login to the local mysql db "
        . "(with limited privs): ";
    $mysql_query_id = trim(fgets(STDIN,256));
    print "mysql pw: ";
    $mysql_query_pw = trim(fgets(STDIN,256));
} while( ($mysql_link = mysql_connect($mysql_host, $mysql_query_id, $mysql_query_pw))
    == FALSE);
mysql_close($mysql_link);

do{
    print "Please enter the pgp id (please make sure the pgp keyrings"
        . "(pub & sec) are in ~/.pgp/, per default): ";
    $pgp_id = trim(fgets(STDIN,256));
}while(strlen($pgp_id) < 1);
do{
    print "pgp pw: ";
    $pgp_pw = trim(fgets(STDIN,256));
} while( strlen($pgp_pw) < 1);

do{
    print "Please enter directory where website is located on server \n" .
        "\t(apache2 default: /usr/local/apache2/htdocs/, \n" .
        "\tinclud slash at end, please): ";
    $webdir = trim(fgets(STDIN,256));
} while( strlen($webdir) < 1);

$my_pub_key_file = $tmpdir . "my_pub_key";

```

```

system("pgp +batchmode +force -kxa $pgp_id $my_pub_key_file");
$my_pub_key = file_get_contents($my_pub_key_file);
$my_pub_key_bin = asc2bin($my_pub_key);
$packet = "n::" . $me . "::" . $pgp_id . "::" . $my_pub_key_bin . "\n";
$socket_n = my_connect($intro, $port_1f);

if( socket_write ($socket_n, $packet, strlen($packet) ) == FALSE ) {
    print socket_strerror( socket_last_error() ) . "\n";
}
if( ($reply_packet = socket_read( $socket_n, $buf, PHP_NORMAL_READ)) == FALSE) {
    print socket_strerror( socket_last_error() ) . "\n";
}
if(preg_match("/e::(.*)/", $reply_packet, $matches) > 0) {
    print $matches[1];
    exit;
}
list($intro_pgp_id, $intro_pub_key_bin, $prev_neighbor, $prev_pgp_id, $prev_pub_key_bin)
    = explode("::", $reply_packet);

$intro_pub_key = bin2asc($intro_pub_key_bin);
$intro_pgp_file = $tmpdir . "intro_pubkey";
my_file_write($intro_pgp_file, $intro_pub_key, "n");
system("pgp +batchmode +force -kat $intro_pgp_file");

$mysql_link = mysql_connect($mysql_host, $mysql_id, $mysql_pw);

/*selecting database*/
if( mysql_select_db($mysql_db) == FALSE) {
    print "could not select database: $mysql_db\n";
}

$sql_pgp_ud = "INSERT INTO pgp VALUES(\"$intro\")";
if( ($mysql_result_ud = mysql_query($sql_pgp_ud) ) == FALSE) {
    print "PGP update failed: " . mysql_error();
}

$prev_pub_key = bin2asc($prev_pub_key_bin);
$prev_pgp_file = $tmpdir . "prev_pubkey";
my_file_write($prev_pgp_file, $prev_pub_key, "n");
system("pgp +batchmode +force -kat $prev_pgp_file");

```

```

$sql_pgp_ud = "INSERT INTO pgp VALUES(\"$prev_neighbor\")";
if( ($mysql_result_ud = mysql_query($sql_pgp_ud) ) == FALSE) {
    print "PGP update failed: " . mysql_error();
}

$sql_pgp_ud = "INSERT INTO pgp VALUES(\"$me\")";
if( ($mysql_result_ud = mysql_query($sql_pgp_ud) ) == FALSE) {
    print "PGP update failed: " . mysql_error();
}

system("cp -f ~/.pgp/pubring.pkr " . $webdir . "pubring.pkr; chmod 444 " .
    $webdir . "pubring.pkr");

socket_close($socket_n);

$config_data = $prev_neighbor . "::<" . $prev_pgp_id . "\n" . $me . "::<" . $pgp_id . "::<" .
    $pgp_pw . "\n" . $intro . "::<" . $intro_pgp_id . "\n" . $mysql_id . "::<" .
    $mysql_pw . "\n" . $mysql_query_id . "::<" . $mysql_query_pw . "\n" . $webdir
    . "\n";
my_file_write("config.dat", $config_data, "n");
$config = file("config.dat");
system("pgp -ew config.dat $pgp_id");

$webconf = $webdir . "config.dat";
$pub_config_data = $prev_neighbor . "\n" . $me . "::<" . $pgp_id . "\n" . $intro . "\n" .
    $mysql_query_id . "::<" . $mysql_query_pw . "\n" . $webdir . "\n";
my_file_write($webconf, $pub_config_data, "n");
system("chmod 444 $webconf");

print "due to the nature of this system, i would like to be able to have some idea " .
    "of the pervasiveness of this project. please type \"y\" to allow an email " .
    "to be sent to robhaining@hotpop.com: ";
$yn = trim(fgets(STDIN,256));

if($yn == "y") {
    print "if you would like to, you may type your email here, and receive " .
        "potential updates/news as they occur: ";
    $email = trim(fgets(STDIN,256));

    if( mail("robhaining@hotpop.com", "new server on the ring", "$me has added " .
        " itself to the ring.\n email: $email\n") ) {

```

```
        print "mail was sent successfully...";
    } else {
        print "mail failed. please manually send a message to "
            . "robhaining@hotpop.com with your radio station name! thanks\n";
    }
}

print "please start server.php to see if the changes have taken effect...\n";
}

?>
```

SERVER.PHP:

```
#!/usr/local/bin/php -q

<?php
/*developed using php 4.3.2*/
    include 'funcs.php';
    error_reporting (E_ALL);

    /*set the time limit to infinity, so we never stop waiting for a connection*/
    set_time_limit(0);

    /* turn on implicit output flushing so we see what we're getting as it comes in */
    ob_implicit_flush();

    $port_lf = 1108; //port for listening for connections, and forwarding on packet to $next_neighbor
    $port_k = 1208;

    $tmpdir = "/tmp/ring/";
    $log = $tmpdir . "log";
    $del_cmd = "rm -f " . $tmpdir . "*";
    system($del_cmd);

    $buf = 32000;

    /*MySQL info*/
    $mysql_db = "localradio";
    $mysql_host = "localhost";
    $mysql_delim = "::"; //mysql delimiter

    if(file_exists("config.dat.pgp")) {
        print "\n\n***After it says \"You need a pass phrase to unlock your secret key,\"
            please enter your PGP passphrase. Thanks!\n\n\n";
        system("pgp config.dat.pgp");
    } else {
        print "the file \"config.dat.pgp\" does not exist in this folder.  if you have not run
            ./config.php, please do.  otherwise, please locate the file \"config.dat.pgp\", and
            return it to this folder. thanks!\n";
        exit;
    }
}
```

```

if( ($config = file("config.dat")) == FALSE) {
    print "there was an error decrypting the config file -- please try restarting server.php\n";
    exit;
}
list($prev_neighbor, $prev_pgp_id) = explode("::", trim($config[0]) );
list($me, $pgp_id, $pgp_pass) = explode("::", trim($config[1]) );
list($next_neighbor, $next_pgp_id) = explode("::", trim($config[2]) );
list($mysql_id, $mysql_pw) = explode("::", trim($config[3]) );
list($mysql_query_id, $mysql_query_pw) = explode("::", trim($config[4]) );
$webdir = trim($config[5]);
system("pgp -w config.dat");

$webconf = $webdir . "config.dat";
$pub_config_data = $prev_neighbor . "\n" . $me . "::" . $pgp_id . "\n" . $next_neighbor . "\n" .
    $mysql_query_id . "::" . $mysql_query_pw . "\n" . $webdir . "\n";
my_file_write($webconf, $pub_config_data, "n");
system("chmod 444 $webconf");

/*connecting to mysql*/
if( ($mysql_link = mysql_connect($mysql_host, $mysql_id, $mysql_pw)) == FALSE) {
    print "Could not connect : " . mysql_error();
}
/*selecting database*/
if( mysql_select_db($mysql_db) == FALSE) {
    print "could not select database: $mysql_db\n";
}

system("cp -f ~/.pgp/pubring.pkr " . $webdir . "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

$socket_l = my_listen($me, $port_lf);

do {
    /* Accept incoming connections: */
    if( ($socket_r = socket_accept($socket_l )) == FALSE) {
        print socket_strerror( socket_last_error() ) . "\n";
        break;
    }

    if( ($packet = socket_read( $socket_r, $buf, PHP_NORMAL_READ)) == FALSE) {
        print socket_strerror( socket_last_error() ) . "\n";
    }
}

```



```

}

if( preg_match("/([\w])::(.*)/", $packet, $matches) == 0) {
    print $packet;
}
$activity = $matches[1];
$remainder = $matches[2];

$data = FALSE;
$query = FALSE;

switch($activity) {
    case "q": //received a query
        socket_getpeername($socket_r, $remote_server_ip);
        $prev_neighbor_ip = gethostbyname($prev_neighbor);
        if($remote_server_ip != $prev_neighbor_ip) {
            socket_close($socket_r);
            print "error...previous neighbor not really the previous
                neighbor?!?\n";
            break;
        }

        socket_close($socket_r);

        $send = false;

list($origin, $sql_cmd, $results_count, $server_count, $orig_pgp_id, $orig_pub_key_bin) =
    explode(":::", $remainder);

    $origin = trim($origin);
    $sql_cmd = trim($sql_cmd);
    $results_count = intval(trim($results_count));
    $server_count = intval(trim($server_count));
    $orig_pgp_id = trim($orig_pgp_id);

    $orig_pub_key = bin2asc(trim($orig_pub_key_bin));

    /*performing sql query*/
    if( ($mysql_result = mysql_query($sql_cmd) ) == FALSE) {
        print "Query failed : " . mysql_error();
    }
}

```

```

}
$hits = mysql_num_rows($mysql_result);

/*update counts..*/
if($server_count > 0) {
    $server_count = $server_count - 1;
    if($server_count == 0) {
        $end = TRUE;
    }
}
if($results_count > 0) {
    $results_count = $results_count - $hits;
    if($results_count <= 0) {
        $end = TRUE;
    }
}
if( gethostbyname($origin) == gethostbyname($next_neighbor) ) {
    $end = TRUE;
}

$senddata = FALSE;
$text = "";
if($hits > 0) {
    /*returns text string of result, delimited by $mysql_delim*/
    while( $line = mysql_fetch_array($mysql_result, MYSQL_ASSOC)) {
        if( preg_match("/$origin/", $line['known_copies']) == 0) {

            foreach($line as $col_value) {
                $text = $text . $col_value . $mysql_delim;
            }
            $text = $text . "\n";

            $senddata = TRUE;
            $copies_ud = $origin . ";" . $line['known_copies'];
            $email = $line['email'];
            $date_pub = $line['date_pub'];
            $time_pub = $line['time_pub'];
            $sql_ud_copies = "UPDATE reviews
                SET known_copies=\"$copies_ud\"
                WHERE email = \"$email\"
                AND date_pub = \"$date_pub\"";

```

```

                                AND time_pub = \"$time_pub\"";
                                if( mysql_query($sql_ud_copies) == FALSE) {
                                    print "update to field \"known copies\"
                                                failed: " . mysql_error();
                                }
                            }
                        }
                    }/*end if($hits > 0)*/

/*Free result set */
mysql_free_result($mysql_result);

if($senddata) {
    /*do i have a public key for $origin?...if not, add it!*/
    $file_key_results = $tmpdir . "pub_key_results";
    system("pgp +batchmode +force -kv $orig_pgp_id > $file_key_results");
    $pub_key_results = file_get_contents($file_key_results);

    preg_match("/\n(\d) matching key/", $pub_key_results, $matches);

    if( intval($matches[1]) == 0 ) {
        /*$orig_pub_key is not already in keyring*/
        /*add origin's public key to keyring ($orig_pub_key)*/
        $tmp_pubkey_file = $tmpdir . "pubkey.asc";
        my_file_write($tmp_pubkey_file, $orig_pub_key, "n");
        system("pgp +batchmode +force -kat $tmp_pubkey_file");
        system("cp -f ~/.pgp/pubring.pkr " . $webdir .
                "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");
    }
    /*put $text in /tmp/ring_text*/
    $tmp_ring_file = $tmpdir . "ring_text";
    my_file_write($tmp_ring_file, $text, "n");

    /*encrypt & sign data => $cipher_text*/
    $tmp_cipher_file = $tmpdir . "ring_text.pgp";

    system("pgp +batchmode +force -es $tmp_ring_file $orig_pgp_id -u" .
            " $pgp_id -o $tmp_cipher_file -z $pgp_pass");
    $cipher_text = file_get_contents($tmp_cipher_file);
}

```

```

/*send data to origin*/
$new_packet = "d::" . $me . "::" . $pgp_id . "\n";

$socket_d = my_connect($origin, $port_1f);
if($socket_d == FALSE) {
    break;
}
if( socket_write ($socket_d, $new_packet, strlen($new_packet) )
    == FALSE ) {
    print socket_strerror( socket_last_error() ) . "\n";
    break;
}

/*binary send! =) */
if( socket_write ($socket_d, $cipher_text, strlen($cipher_text) )
    == FALSE ) {
    print socket_strerror( socket_last_error() ) . "\n";
    break;
}

socket_close($socket_d);

/*if that server does not have your public key,
send it to the next neighbor*/
$sql_pgp = "SELECT * FROM pgp WHERE server=\"$origin\"";

if( ($mysql_result = mysql_query($sql_pgp) ) == FALSE) {
    print "PGP query failed : " . mysql_error();
}
if( mysql_num_rows($mysql_result) == 0 ) {
    /*extract my signed public key*/
    $pubkey_file = $tmpdir . "pubkey.asc";
    system("pgp +batchmode +force -kxa $pgp_id $pubkey_file");
    system("pgp +batchmode +force -sta $pubkey_file " .
        "-u $pgp_id -z $pgp_pass");
    $my_pubkey = file_get_contents($pubkey_file);
    $my_pubkey_bin = asc2bin($my_pubkey);

    $packet = "k::" . $origin . "::" . $pgp_id . "::" .
        $my_pubkey_bin . "\n";
}

```

```

$self_heal = FALSE;
if($origin == $next_neighbor) {
    if( my_socket_write($next_neighbor, $port_k, $packet)
        == FALSE) {
        $self_heal = TRUE;
    }
} else {
    if( my_socket_write($next_neighbor, $port_lf, $packet)
        == FALSE) {
        $self_heal = TRUE;
    }
}
}
/**SELF-HEALING-PROCESS INVOKED***/
if($self_heal) {
    $packet = "s::p::" . $me . "::" . $pgp_id . "::" .
        $my_pubkey_bin . "\n";
    my_socket_write($prev_neighbor, $port_lf, $packet);

    $packet_o = "e::The ring is temporarily broken. " .
        "Please try back again later on.\n";
    my_socket_write($origin, $port_lf, $packet_o);
}

$sql_pgp_ud = "INSERT INTO pgp VALUES(\"$origin\")";
if( ($mysql_result_ud = mysql_query($sql_pgp_ud) ) == FALSE) {
    print "PGP update failed: " . mysql_error();
}
} /*end if(mysql_num_rows($mysql_result) == 0) */
/*Free result set */
mysql_free_result($mysql_result);

} /*end if($senddata)*/

if($end) {
    /*send stop listening signal to origin*/
    $new_packet = "x::\n";
    my_socket_write($origin, $port_lf, $new_packet);
} else {
    $orig_pubkey_file = $tmpdir . "orig_pubkey.asc";
    my_file_write($orig_pubkey_file, $orig_pub_key, "n");
}

```

```

system("pgp +batchmode +force -kat $orig_pubkey_file");
system("cp -f ~/.pgp/pubring.pkr " . $webdir .
      "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

/*sign public key*/
$sig_results = $tmpdir . "sig_results";
system("pgp +batchmode +force -kvv $orig_pgp_id > $sig_results");
$sigs = file_get_contents($sig_results);
if( preg_match("/$pgp_id/", $sigs) == 0) {
    system("pgp +batchmode +force -ks $orig_pgp_id" .
          " -u $pgp_id -z $pgp_pass");
}

/*extract signed public key*/
$pubkey_file = $tmpdir . "pubkey.asc";
system("pgp +batchmode +force -kxa $orig_pgp_id $pubkey_file");
$orig_pub_key = file_get_contents($pubkey_file);
$orig_pub_key_bin = asc2bin($orig_pub_key);

/*forward request along to next neighbor*/
$new_packet = "q::" . $origin . "::" . $sql_cmd . "::" .
             $server_count . "::" . $results_count . "::" .
             $orig_pgp_id . "::" . $orig_pub_key_bin;
$new_packet = trim($new_packet) . "\n";
if(my_socket_write($next_neighbor, $port_lf, $new_packet) == FALSE) {
    $pubkey_file = $tmpdir . "pubkey.asc";
    system("pgp +batchmode +force -kxa $pgp_id $pubkey_file");
    system("pgp +batchmode +force -sta $pubkey_file " .
          "-u $pgp_id -z $pgp_pass");
    $my_pubkey = file_get_contents($pubkey_file);
    $my_pubkey_bin = asc2bin($my_pubkey);

/**SELF-HEALING-PROCESS INVOKED***/

    $packet = "s::p::" . $me . "::" . $pgp_id . "::" .
             $my_pubkey_bin . "\n";
    my_socket_write($prev_neighbor, $port_lf, $packet);

    $packet_o = "e::The ring is temporarily broken." .
               " Please try back again later on.\n";
    my_socket_write($origin, $port_lf, $packet_o);
}

```

```

    }
}

break;

case "u": /*update!*/
    socket_close($socket_r);

    $sql_ud = trim(bin2asc($remainder));

    preg_match("/date_mod=\"(\d{4}-\d{2}-\d{2})\",time_mod=\"\" .
        \"(\d{2}:\d{2}:\d{2})\".*WHERE email=\"(.*@.*)\", AND \" .
        \"date_pub=\"(\d{4}-\d{2}-\d{2})\" AND \" .
        \"time_pub=\"(\d{2}:\d{2}:\d{2})\"/\"", $sql_ud, $matches);

    $date_pub = $matches[4];
    $time_pub = $matches[5];
    $date_mod = $matches[1];
    $time_mod = $matches[2];
    $email = $matches[3];

    $sql_query = "SELECT * FROM reviews WHERE date_pub=\"$date_pub\" .
        \" AND time_pub=\"$time_pub\" AND email=\"$email\"";
    if( ($mysql_result = mysql_query($sql_query) ) == FALSE) {
        print "Query failed : " . mysql_error();
    }

    $new_data = TRUE;
    while( $row = mysql_fetch_array($mysql_result, MYSQL_ASSOC) ) {
        if( ($row['date_mod'] == $date_mod)
            && ($row['time_mod'] == $time_mod) ) {
            $new_data = FALSE;
            break;
        }
    }

    mysql_free_result($mysql_result);

    if($new_data) {
        /*performing local sql update*/
        if( mysql_query($sql_ud) == FALSE) {

```

```

        print "Update failed : " . mysql_error();
    }

    /*FORWARD UPDATE to servers listed under "known_copies"*/
    preg_match("/(WHERE.*)/", $sql_ud, $matches);
    $sql_known_query = "SELECT known_copies FROM reviews " . $matches[1];

    /*performing sql query*/
    if( ($mysql_result = mysql_query($sql_known_query) ) == FALSE) {
        print "Query failed : " . mysql_error();
    }

    $servers = mysql_fetch_array($mysql_result, MYSQL_ASSOC);
    $server_arr = explode(";", $servers['known_copies']);

    foreach($server_arr as $server) {
        if(strlen($server) > 1) {
            my_socket_write($server, $port_lf, $packet);
        }
    }

    mysql_free_result($mysql_result);
}/*end if($new_data)*/

break;

case "f": //received a flush command
    socket_close($socket_r);

    $sql_del = trim(bin2asc($remainder));

    /*FORWARD UPDATE to servers listed under "known_copies"*/
    preg_match("/(WHERE.*)/", $sql_del, $matches);
    $sql_known_query = "SELECT known_copies FROM reviews " . $matches[1];

    /*performing sql query*/
    if( ($mysql_result = mysql_query($sql_known_query) ) == FALSE) {
        print "Query failed : " . mysql_error();
    }

    $servers = mysql_fetch_array($mysql_result, MYSQL_ASSOC);

```



```

$server_arr = explode(";", $servers['known_copies']);

/*performing local sql delete*/
if( mysql_query($sql_del) == FALSE) {
    print "Delete failed : " . mysql_error();
}

foreach($server_arr as $server) {
    if(strlen($server) > 1) {
        my_socket_write($server, $port_lf, $packet);
    }
}

mysql_free_result($mysql_result);

break;

case "d": //received data
    if(($cipher_text = socket_read( $socket_r, $buf, PHP_BINARY_READ)) == FALSE) {
        print socket_strerror( socket_last_error() ) . "\n";
        break;
    }
    socket_close($socket_r);

    $have_key = FALSE;
    $socket_kl = my_listen($me, $port_k);

    while($have_key == FALSE) {

        list($home, $home_pgp_id) = explode($mysql_delim, $remainder);
        $home = trim($home);
        $home_pgp_id = trim($home_pgp_id);

        /*do i have a pub key for $home?....
        if not, listen on the socket to wait for one*/
        $file_key_results = $tmpdir . "pub_key_results";
        system("pgp +batchmode +force -kvv $home_pgp_id >" .
            " $file_key_results");
        $pub_key_results = file_get_contents($file_key_results);

        preg_match("/(\d) matching key/", $pub_key_results, $matches);

```

```

if( intval($matches[1]) > 0) {
    $have_key = TRUE;
    break;
}

/* Accept incoming connections: */
if( ($socket_k = socket_accept($socket_kl )) == FALSE) {
    print socket_strerror( socket_last_error() ) . "\n";
    break;
}

/* Read/Receive some data from the client: */
if( ($packet_key = socket_read( $socket_k, $buf, PHP_NORMAL_READ))
    == FALSE) {
    print socket_strerror( socket_last_error() ) . "\n";
    break;
} else {$have_key = TRUE;}
socket_close($socket_k);

if( preg_match("/e::(.*)/", $packet_key, $matches) > 0) {
    print $matches[1];
    break;
}

/*authenticate signature, then add key to public key ring*/
list($letter, $orig_server, $home_pgp_id, $key_sig_bin) =
    explode("::", $packet_key);
$key_sig = bin2asc($key_sig_bin);
$key_file_sig = $tmpdir . "key_sig";
$key_file = $tmpdir . "key";
my_file_write($key_file_sig, $key_sig, "n");

system("pgp +batchmode $key_file_sig -o $key_file", $result);
if($result == 0) {
    /*good key, add to public key ring*/
    system("pgp +batchmode +force -kat $key_file", $result);
    system("cp -f ~/.pgp/pubring.pkr " . $webdir .
        "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");
    break;
} else {
    /*bad key, send msg back to prev neighbor*/
    $have_key = FALSE;
}

```

```

    }

}
socket_close($socket_kl);

/*once it's received, authenticate and decrypt the data*/
$cipher_file = $tmpdir . "cipher_file";
my_file_write($cipher_file, $cipher_text, "b");

$tmp_data_file = $tmpdir . "data_table";
system("pgp +batchmode +force $cipher_file -z $pgp_pass -o $tmp_data_file");
$data = file($tmp_data_file);

/*INSERT DATA INTO DB*/
foreach($data as $row) {
    list($author,$email,$affiliate,$date_pub,$time_pub,$date_mod,
        $time_mod,$origin,$known_copies,$topic,$genre,$website,$label,
        $contact,$review,$null_tmp) = explode($mysql_delim, $row);
    $sql_check = "SELECT *
    FROM reviews
    WHERE email = \"$email\"
        and date_pub = \"$date_pub\"
        and time_pub = \"$time_pub\"";
    $sql_insert = "INSERT INTO reviews
    VALUES (\"$author\", \"$email\", \"$affiliate\",
        \"$date_pub\", \"$time_pub\", \"$date_mod\",
        \"$time_mod\", \"$origin\", \"\", \"$topic\",
        \"$genre\", \"$website\", \"$label\",
        \"$contact\", \"$review\")";
    if( ($mysql_result = mysql_query($sql_check) ) == FALSE) {
        print "Query failed : " . mysql_error();
    } else {
        if(mysql_num_rows($mysql_result) == 0) {
            if( mysql_db_query($mysql_db, $sql_insert) == FALSE) {
                print "Insert failed : " . mysql_error();
            }
        }
        /*end if(mysql_num_rows($mysql_result) == 0)*/
    }
    /*end if&else ( ($mysql_result = mysql_query($sql_check) )
    == FALSE)*/
    mysql_free_result($mysql_result);
}/*end foreach($data_rows_q as $row)*/

```

```

        break;

case "x": //received a stop listening command
    socket_close($socket_r);

    if(strlen($remainder) > 1) {
        print "received this packet: $remainder\n";
    }

    /*shared memory stuff*/
    $MEMSIZE = 512;
    $SHMKEY = 2;

    if( ($shm_id = shm_attach($SHMKEY, $MEMSIZE)) == FALSE) {
        print "failed to attach shared memory\n";
    }
    shm_put_var($shm_id, 1, "<stop_listening\0>");
    /*end shared memory stuff*/
    break;

case "k": /*received a key...need to authenticate, remove sig,
          sign, and fwd to next neighbor*/
    socket_close($socket_r);

    list($origin, $home_pgp_id, $home_pubkey_bin) = explode("::", $remainder);
    $origin = trim($origin);
    $home_pgp_id = trim($home_pgp_id);
    $home_pubkey = trim(bin2asc($home_pubkey_bin));
    $home_pubkey_file_sig = $tmpdir . "home_pubkey_sig.asc";
    $home_pubkey_file = $tmpdir . "home_pubkey.asc";
    my_file_write($home_pubkey_file_sig, $home_pubkey, "n");

    system("pgp +batchmode $home_pubkey_file_sig -o $home_pubkey_file", $result);
    system("cp -f ~/.pgp/pubring.pkr " . $webdir .
           "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

    if($result == 0) {
        /*good signature file...sign and forward onto next neighbor*/
        system("pgp +batchmode +force -kat $home_pubkey_file");
        system("cp -f ~/.pgp/pubring.pkr " . $webdir .

```

```

        "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

/*sign public key*/
$sig_results = $tmpdir . "sig_results";
system("pgp +batchmode +force -kvv $home_pgp_id > $sig_results");
$sigs = file_get_contents($sig_results);
if( preg_match("/$pgp_id/", $sigs) == 0) {
    system("pgp +batchmode +force -ks $home_pgp_id" .
        " -u $pgp_id -z $pgp_pass");
}

/*extract signed public key*/
$new_pubkey_file = $tmpdir . "new_pubkey.asc";
system("pgp +batchmode +force -kxa $home_pgp_id $new_pubkey_file");
system("pgp +batchmode +force -sta $new_pubkey_file" .
    " -u $pgp_id -z $pgp_pass");
$new_pubkey = file_get_contents($new_pubkey_file);
$new_pubkey_bin = asc2bin($new_pubkey);

$key_packet = "k::" . $origin . "::" . $home_pgp_id . "::" .
    $new_pubkey_bin . "\n";

$self_heal = FALSE;
if($next_neighbor == $origin) {
    if( my_socket_write($origin, $port_k, $key_packet) == FALSE) {
        $self_heal = TRUE;
    }
} else {
    if( my_socket_write($next_neighbor, $port_lf, $key_packet)
        == FALSE) {
        $self_heal = TRUE;
    }
}
if($self_heal) {
    $my_pgp_file = $tmpdir . "my_pub_key";
    system("pgp +batchmode +force -kxa $pgp_id $my_pgp_file");
    $my_pub_key = file_get_contents($my_pgp_file);
    $my_pub_key_bin = asc2bin($my_pub_key);

    /**SELF-HEALING-PROCESS INVOKED***/
    $packet = "s::p::" . $me . "::" . $pgp_id . "::" .

```

```

        $my_pub_key_bin . "\n";
my_socket_write($prev_neighbor, $port_lf, $packet);

$packet_o = "e::The ring is temporarily broken." .
    " Please try back again later on.\n";
my_socket_write($origin, $port_lf, $packet_o);
    }

} else {
    print "error: could not get a good reading from the pgp key...\n";
}

break;

case "n": /*there's a new server in town! let's add, baby*/

list($new_server, $new_pgp_id, $new_pgp_key_bin) = explode("::", $remainder);
$new_pgp_key = bin2asc($new_pgp_key_bin);
$new_pgp_file = $tmpdir . "new_pgp_key";
my_file_write($new_pgp_file, $new_pgp_key, "n");
system("pgp +batchmode +force -kat $new_pgp_file");
system("pgp +batchmode +force -ks $new_pgp_id -u $pgp_id -z $pgp_pass");
system("cp -f ~/.pgp/pubring.pkr " . $webdir .
    "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

$sql_pgp_ins = "INSERT INTO pgp VALUES(\"$new_server\")";
if( mysql_query($sql_pgp_ins) == FALSE) {
    print "PGP update failed: " . mysql_error();
}

$my_pgp_file = $tmpdir . "my_pub_key";
system("pgp +batchmode +force -kxa $pgp_id $my_pgp_file");
$my_pub_key = file_get_contents($my_pgp_file);
$my_pub_key_bin = asc2bin($my_pub_key);

$prev_pgp_file = $tmpdir . "prev_pub_key";
system("pgp +batchmode +force -kxa $prev_pgp_id $prev_pgp_file");
$prev_pub_key = file_get_contents($prev_pgp_file);
$prev_pub_key_bin = asc2bin($prev_pub_key);

$reply = $pgp_id . "::" . $my_pub_key_bin . "::" . $prev_neighbor . "::" .

```

```

        $prev_pgp_id . ":@" . $prev_pub_key_bin . "\n";

if( socket_write ($socket_r, $reply, strlen($reply) ) == FALSE ) {
    print socket_strerror( socket_last_error() ) . "\n";
}

$new_packet = "i:@" . trim($remainder) . "\n";
if( my_socket_write($prev_neighbor, $port_lf, $new_packet) == FALSE) {
/**SELF-HEALING-PROCESS INVOKED***/
    $packet = "s:n:@" . $me . ":@" . $pgp_id . ":@" .
        $my_pub_key_bin . "\n";
    my_socket_write($next_neighbor, $port_lf, $packet);

    $packet_o = "e::The ring is temporarily broken. " .
        "Please try back again later on.\n";
    if( socket_write ($socket_r, $packet_o, strlen($packet_o) )
        == FALSE ) {
        print socket_strerror( socket_last_error() ) . "\n";
    }
}
socket_close($socket_r);

system("pgp +batchmode +force config.dat.pgp -z $pgp_pass");
$prev_neighbor = $new_server;
$prev_pgp_id = $new_pgp_id;
$config = file("config.dat");
$config[0] = $new_server . ":@" . $new_pgp_id . "\n";
$config_str = "";
foreach($config as $line) {
    $config_str .= $line;
}
/*print "config data:\n$config_str\n";*/
my_file_write("config.dat", $config_str, "n");
system("pgp +batchmode +force -ew config.dat $pgp_id");

$pub_config_data = $prev_neighbor . "\n" . $me . ":@" . $pgp_id . "\n" .
    $next_neighbor . "\n" . $mysql_query_id . ":@" .
    $mysql_query_pw . "\n" . $webdir . "\n";
$webconf = $webdir . "config.dat";
my_file_write($webconf, $pub_config_data, "n");
system("chmod 444 $webconf");

```

```

        break;

case "i":
    socket_close($socket_r);

    list($new_server, $new_pgp_id, $new_pub_key_bin) = explode("::", $remainder);
    $new_pub_key = bin2asc($new_pub_key_bin);
    $new_pub_file = $tmpdir . "new_pub_key";
    my_file_write($new_pub_file, $new_pub_key, "n");
    system("pgp +batchmode +force -kat $new_pub_file");
    system("pgp +batchmode +force -ks $new_pgp_id -u $pgp_id -z $pgp_pass");
    $sql_pgp_ud = "INSERT INTO pgp VALUES(\"$new_server\")";
    system("cp -f ~/.pgp/pubring.pkr " . $webdir .
           "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

    if( ($mysql_result_ud = mysql_query($sql_pgp_ud) ) == FALSE) {
        print "PGP update failed: " . mysql_error();
    }

    system("pgp +batchmode +force config.dat.pgp -z $pgp_pass");
    $next_neighbor = $new_server;
    $config = file("config.dat");
    $config[2] = $next_neighbor . "::" . $new_pgp_id . "\n";
    $config_str = "";
    foreach($config as $line) {
        $config_str .= $line;
    }
    my_file_write("config.dat", $config_str, "n");
    system("pgp +batchmode +force -ew config.dat $pgp_id");

    $pub_config_data = $prev_neighbor . "\n" . $me . "::" . $pgp_id . "\n" .
                      $next_neighbor . "\n" . $mysql_query_id . "::" .
                      $mysql_query_pw . "\n" . $webdir . "\n";
    $webconf = $webdir . "config.dat";
    my_file_write($webconf, $pub_config_data, "n");
    system("chmod 444 $webconf");

    break;

case "s":

```



```

socket_close($socket_r);
/**SELF-HEALING-PROCESS: case s****/
list($which_neigh, $home, $home_pgp_id, $home_pubkey_bin)
    = explode("::", $remainder);
$which_neigh = trim($which_neigh);
$home = trim($home);
$home_pgp_id = trim($home_pgp_id);
$home_pubkey_bin = trim($home_pubkey_bin);

if($which_neigh == "n") {
    $neighbor1 = $next_neighbor;
    $neighbor2 = $prev_neighbor;
    $h_neigh = "p";
} else if($which_neigh == "p") {
    $neighbor1 = $prev_neighbor;
    $neighbor2 = $next_neighbor;
    $h_neigh = "n";
}

if( my_socket_write($neighbor1, $port_lf, $packet) == FALSE) {
    /*change config file!*/
    $home_pub_key = bin2asc($home_pubkey_bin);
    $home_pub_file = $tmpdir . "home_pub_key";
    my_file_write($home_pub_file, $home_pub_key, "n");
    system("pgp +batchmode +force $home_pub_file");
    system("pgp +batchmode +force -kat $home_pub_file");
    system("pgp +batchmode +force -ks $home_pgp_id " .
        "-u $pgp_id -z $pgp_pass");
    system("cp -f ~/.pgp/pubring.pkr " . $webdir .
        "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

    system("pgp +batchmode +force config.dat.pgp -z $pgp_pass");
    $config = file("config.dat");

    if($which_neigh == "n") {
        $next_neighbor = $home;
        $config[2] = $next_neighbor . "::" . $home_pgp_id . "\n";
    } else if($which_neigh == "p") {
        $prev_neighbor = $home;
        $config[0] = $prev_neighbor . "::" . $home_pgp_id . "\n";
    }
}

```

```

$config_str = "";
foreach($config as $line) {
    $config_str .= $line;
}
my_file_write("config.dat", $config_str, "n");
system("pgp +batchmode +force -ew config.dat $pgp_id -z $pgp_pass");

$pub_config_data = $prev_neighbor . "\n" . $me . ":@" . $pgp_id .
    "\n" . $next_neighbor . "\n" . $mysql_query_id . ":@"
    . $mysql_query_pw . "\n" . $webdir . "\n";
$webconf = $webdir . "config.dat";
my_file_write($webconf, $pub_config_data, "n");
system("chmod 444 $webconf");

$my_pgp_file = $tmpdir . "my_pub_key";
system("pgp +batchmode +force -kxa $pgp_id $my_pgp_file");
$my_pub_key = file_get_contents($my_pgp_file);
$my_pub_key_bin = asc2bin($my_pub_key);

$new_packet = "h:@" . $h_neigh . ":@" . $home . ":@" . $me . ":@" .
    $pgp_id . ":@" . $my_pub_key_bin . "\n";
my_socket_write($neighbor2, $port_lf, $new_packet);
}/*end if/else (socket_connect..)*/

break;

case "h":
    /***SELF-HEALING-PROCESS: case h***/
    socket_close($socket_r);
    list($which_neigh, $server, $home, $home_pgp_id, $home_pubkey_bin)
        = explode(":", $remainder);
    $which_neigh = trim($which_neigh);
    $server = trim($server);
    $home = trim($home);
    $home_pgp_id = trim($home_pgp_id);
    $home_pubkey_bin = trim($home_pubkey_bin);

    if($which_neigh == "n") {
        $neighbor = $next_neighbor;
    } else if($which_neigh == "p") {
        $neighbor = $prev_neighbor;
    }

```

```

}

if( ($server == $me) || (my_socket_write($neighbor, $port_lf, $packet)
== FALSE) ) {
    $new_pub_key = bin2asc($home_pubkey_bin);
    $new_pub_file = $tmpdir . "new_pub_key";
    my_file_write($new_pub_file, $new_pub_key, "n");
    system("pgp +batchmode +force $new_pub_file");
    system("pgp +batchmode +force -kat $new_pub_file");
    system("pgp +batchmode +force -ks $new_pgp_id"
        . " -u $pgp_id -z $pgp_pass");
    system("cp -f ~/.pgp/pubring.pkr " . $webdir .
        "pubring.pkr; chmod 444 " . $webdir . "pubring.pkr");

    /*change config file!*/
    system("pgp +batchmode +force config.dat.pgp -z $pgp_pass");
    $config = file("config.dat");
    system("pgp -w config.dat");

    if($which_neigh == "n") {
        $next_neighbor = $home;
        $config[2] = $next_neighbor . "::" . $pgp_id . "\n";
    } else if($which_neigh == "p") {
        $prev_neighbor = $home;
        $config[0] = $prev_neighbor . "::" . $pgp_id . "\n";
    }
    $config_str = "";
    foreach($config as $line) {
        $config_str .= $line;
    }
    my_file_write("config.dat", $config_str, "n");
    system("pgp +batchmode +force -ew config.dat $pgp_id -z $pgp_pass");

    $pub_config_data = $prev_neighbor . "\n" . $me . "::" . $pgp_id .
        "\n" . $next_neighbor . "\n" . $mysql_query_id . "::"
        . $mysql_query_pw . "\n" . $webdir . "\n";
    $webconf = $webdir . "config.dat";
    my_file_write($webconf, $pub_config_data, "n");
    system("chmod 444 $webconf");
}

```

```
        break;

    case "e":
        socket_close($socket_r);
        print $remainder;

        /*shared memory stuff*/
        $MEMSIZE = 512;
        $SHMKEY = 2;

        if( ($shm_id = shm_attach($SHMKEY, $MEMSIZE)) == FALSE) {
            print "failed to attach shared memory\n";
        }
        shm_put_var($shm_id, 1, "<stop_listening\0>");
        /*end shared memory stuff*/

        break;

    default:
        print "unknown error!!\n";
        socket_close($socket_r);

        break;
    }
} while(true);

/*closing connection*/
mysql_close($mysql_link);

/* Close the global listening socket: */
socket_close($socket_l);
```

?>

WWW SCRIPTS - MAIN.PHP:

```
<html>
<body>

<?php
$mood = $_POST['mood'];
    switch($mood) {
        case "a": include 'add.php'; break;
        case "d": include 'delete.php'; break;
        case "f": include 'flush.php'; break;
        case "q": include 'query.php'; break;
        case "s": include 'show_db.php'; break;
        case "u": include 'update.php'; break;
        default:
            ?>
            <p>Greetings, how many I help you today?</p>

            <form method="post" action="<?php echo $PHP_SELF ?>">
                <select name="mood">
                    <option value="q">Query Ring</option>
                    <option value="s">Show Local DB</option>
                    <option value="a">Add Review*</option>
                    <option value="u">Update Review to Ring*</option>
                    <option value="d">Delete Review from local DB*</option>
                    <option value="f">Flush Review from Ring*</option>
                </select><br>
                *admin only<br>
                <input type="Submit" name="submit" value="submit request">
            </form>
            <?php
            break;
        }
    ?>

</body>
</html>
```

WWW SCRIPTS - QUERY.PHP:

```
<?php
    include "funcs.php";
    include "header.html";

    //error_reporting (E_ALL);
    set_time_limit(0); /*sets time limit for execution time*/

    if( ($config = file("config.dat")) == FALSE) {
        print "couldn't open file: config.dat\n";
        exit();
    }
    $prev_neighbor = trim($config[0]);
    list($me, $pgp_id) = explode("::", trim($config[1]) );
    $next_neighbor = trim($config[2]);
    list($mysql_id, $mysql_pw) = explode("::", trim($config[3]) );
    $webdir = trim($config[4]);

    $port_q = 1108;

    /*MySQL Stuff*/
    $mysql_host = "localhost";
    $mysql_db = "localradio";
    $mysql_table = "reviews";
    $mysql_delim = "::";

    $dir = "/tmp/ring/client/";
    $del_cmd = "rm -f " . $dir . "*";
    system($del_cmd);

    $keyring = $webdir . "pubring.pkr";
    putenv("PUBRING=$keyring");

    if($_POST['action'] != "query_submit") {
        ?>
        <form method="post" action="query.php">
        search by: <select name="search_by">
            <option value="topic">topic (artist/album/...)</option>
            <option value="genre">genre</option>
            <option value="label">label</option>
```

```

        <option value="all">all</option>
    </select>
    ...enter selection: <input type="text" name="selection"><br>
    enter # of servers to hit (0,+): <input type="text" name="server_count"><br>
    enter # of results to hit (0,+): <input type="text" name="result_count"><br>
    <input type="hidden" name="action" value="query_submit">
    <input type="submit" name="submit" value="submit">
    <?php
} else {
    $search_by = $_POST['search_by'];
    $selection = $_POST['selection'];
    if($search_by == "all") {
        $where = "";
    } else {
        $where = "WHERE $search_by REGEXP \"$selection\"";
    }

    $sql_cmd = "SELECT * FROM $mysql_table " . $where . " ORDER BY topic";

    $result_count = $_POST['result_count'];
    $server_count = $_POST['server_count'];

    /*create connection to mysql*/
    if( ($mysql_link = mysql_connect($mysql_host, $mysql_id, $mysql_pw)) == FALSE) {
        print "Could not connect : " . mysql_error();
        exit(1);
    }
    /*connects to specific local database*/
    if( mysql_select_db($mysql_db) == FALSE) {
        print "could not select database: $mysql_db\n";
    }

    $send_query = TRUE;
    if($result_count > 0) {
        /*performing sql query*/
        if( ($mysql_result = mysql_query($sql_cmd) ) == FALSE) {
            print "Query failed : " . mysql_error();
        }
        $num_rows = mysql_num_rows($mysql_result);
        if($num_rows > 0) {$result_count = $result_count - $num_rows; }
        if($result_count < 0) {

```

```

        $send_query = FALSE;
    }
    mysql_free_result($mysql_result);
}

if($send_query) {
    $tmp_pubkey = $dir . "pubkey.asc";
    system("rm -f $tmp_pubkey");

    $log = $dir . "log";
    system("pgp +batchmode +force -kxa $pgp_id $tmp_pubkey $keyring > $log");
    system("pgp +batchmode +force -sta $tmp_pubkey -u $pgp_id -z $pgp_pass");
    $pub_key = file_get_contents($tmp_pubkey);
    $pub_key_bin = asc2bin($pub_key);

    $packet = "q::" . $me . "::" . $sql_cmd . "::" . $result_count . "::" . $server_count
        . "::" . $pgp_id . "::" . $pub_key_bin . "\n";

    if( my_socket_write($next_neighbor, $port_q, $packet) == FALSE) {
        $shp_packet = "s::p::" . $me . "::" . $pgp_id . "::" . $pub_key_bin . "\n";
        my_socket_write($prev_neighbor, $port_q, $shp_packet);

        print "<br><h2>The ring is temporarily broken.
            Please try back again in a few minutes.</h2><br>\n";
        exit;
    }

    /* start shared memory stuffs*/
    $MEMSIZE = 512;
    $SHMKEY = 2;
    $shm_id = shm_attach($SHMKEY,$MEMSIZE);
    if($shm_id == FALSE) {
        print "failed to attach shared memory\n";
        exit();
    }
    if( (shm_put_var($shm_id, 1, "<not_ready>")) == FALSE){
        print "failed to put var1 on shared memory $shm_id\n";
        shm_remove($shm_id);
        exit();
    }
}

```



```

$shm_var = shm_get_var($shm_id,1);

//loop until server.php changes the shared memory to <mysql_is_updated>
while($shm_var == "<not_ready>") {
    $shm_var = shm_get_var($shm_id,1);
}

if(shm_remove($shm_id) == FALSE) {
    print "failed to remove $shm_id from shared memory\n";
}
}/*end if($send_query)*/

if( ($mysql_result = mysql_query($sql_cmd) ) == FALSE) {
    print "Query failed : " . mysql_error();
}
if(mysql_num_rows($mysql_result) > 0) {
    while( $line = mysql_fetch_array($mysql_result, MYSQL_ASSOC) ) {
        ?>
        Topic: <a href="<?php echo $line['website']?>" target="_blank">
            <?php echo $line['topic']?></a><br>
        Author: <a href="mailto:<?php echo $line['email']?>">
            <?php echo $line['author']?></a><br>
        Affiliate Station: <?php echo $line['affiliate']?><br>
        Published: <?php echo $line['date_pub']?><br>
        Last Modified: <?php echo $line['date_mod']?><br>
        Genre: <?php echo $line['genre']?><br>
        Label: <?php echo $line['label']?> ... <?php echo $line['contact']?><br>
        Review: <?php echo $line['review']?></p>
        <?php
    }
}
/*Free result set */
mysql_free_result($mysql_result);
/*closing connection*/
mysql_close($mysql_link);

}
include "footer.html";
?>

```

WWW SCRIPTS - INSERT.PHP:

```
<?php
include "header.html";
//error_reporting (E_ALL);
set_time_limit(0); /*sets time limit for execution time*/

if( ($config = file("config.dat")) == FALSE) {
    print "couldn't open file: config.dat\n";
    exit(1);
}
$prev_neighbor = trim($config[0]);
list($me, $pgp_id) = explode("::", trim($config[1]) );
$next_neighbor = trim($config[2]);
list($mysql_id, $mysql_pw) = explode("::", trim($config[3]) );
$webdir = trim($config[4]);

/*global config stuff*/
$port_q = 1108;

/*MySQL Stuff...check local db first*/
$mysql_host = "localhost";
$mysql_db = "localradio";
$mysql_table = "reviews";
$mysql_delim = "::";

if($HTTP_POST_VARS["action"] == "data_submitted") {
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "your id & pw are not valid. please try again<br>\n";
        exit;
    }
    $date_pub = date("Y-m-d");
    $date_mod = $date_pub;
    $time_pub = date("G:i:s");
    $time_mod = $time_pub;

    /*connects to specific local database*/
    if( mysql_select_db($mysql_db) == FALSE) {
        print "could not select database: $mysql_db\n";
    }
}
```

```

}

$author = $_POST['author'];
$email = $_POST['email'];
$affiliate = $_POST['affiliate'];
$topic = $_POST['topic'];
$genre = $_POST['genre'];
$website = $_POST['website'];
$label = $_POST['label'];
$contact = $_POST['contact'];
$review = $_POST['review'];

$sql_new = "INSERT INTO $mysql_table
            VALUES(\"$author\", \"$email\", \"$affiliate\",
                    \"$date_pub\", \"$time_pub\", \"$date_mod\", \"$time_mod\",
                    \"$me\", \"\", \"$topic\", \"$genre\", \"$website\",
                    \"$label\", \"$contact\", \"$review\")";
if( ( mysql_db_query($mysql_db, $sql_new) ) == FALSE) {
    print "Insert failed : " . mysql_error();
    break;
}
else print "Review was inserted successfully\n";

/*closing connection*/
mysql_close($mysql_link);

} else if($HTTP_POST_VARS["action"] == "login") {
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "your id & pw are not valid. please try again<br>\n";
        exit;
    } else {
        ?>
        <form method="post" action="add.php">
            Welcome...<br>
            enter name: <input type="text" name="author"><br>
            email address: <input type="text" name="email"><br>
            radio station affiliation: <input type="text" name="affiliate"><br>
            artist name/album name/show date/...: <input type="text" name="topic"><br>
            genre: <input type="text" name="genre"><br>
            website for artist: <input type="text" name="website" value="http://"><br>
            current label: <input type="text" name="label"><br>

```

```
        contact info for label: <input type="text" name="contact"><br>
        Review: <textarea name="review" cols=60 rows=15 wrap="wrap"></textarea><br>
        <input type="hidden" name="id" value="<?php echo $_POST['id']?>">
        <input type="hidden" name="pw" value="<?php echo $_POST['pw']?>">
        <input type="hidden" name="action" value="data_submitted">
        <input type="Submit" name="submit2" value="add review">
    </form>
    <?php
    }
} else {
    print "you have chosen to add a review to the local database. "
        "please enter login info...<br>\n";
    ?>
    <form method="post" action="add.php">
        DB id: <input type="Text" name="id"><br>
        password: <input type="password" name="pw"><br>
        <input type="hidden" name="action" value="login">
        <input type="Submit" name="submit1" value="submit info">
    </form>
    <?php
}
include "footer.html";
?>
```

WWW SCRIPTS - DELETE.PHP:

```
<?php
include "header.html";
//error_reporting (E_ALL);
set_time_limit(0); /*sets time limit for execution time*/

if( ($config = file("config.dat")) == FALSE) {
    print "couldn't open file: config.dat\n";
    exit(1);
}
$prev_neighbor = trim($config[0]);
list($me, $pgp_id) = explode("::", trim($config[1]) );
$next_neighbor = trim($config[2]);
list($mysql_id, $mysql_pw) = explode("::", trim($config[3]) );
$webdir = trim($config[4]);

/*global config stuff*/
$port_q = 1108;

/*MySQL Stuff...check local db first*/
$mysql_host = "localhost";
$mysql_db = "localradio";
$mysql_table = "reviews";
$mysql_delim = "::";

if($_POST["action"] == "select_review") {
    /*create connection to mysql*/
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "Could not connect : " . mysql_error();
        break;
    }

    /*connects to specific local database*/
    if( mysql_select_db($mysql_db) == FALSE) {
        print "could not select database: $mysql_db\n";
        break;
    }

    list($date, $time) = explode(";", $_POST['date_time']);
```

```

$email = $_POST['email'];

$sql_del = "DELETE FROM $mysql_table WHERE email=\"\$email\" AND date_pub=\"\$date\" "
          . "AND time_pub=\"\$time\"";

if(mysql_query($sql_del) == FALSE) {
    print "could not delete: " . mysql_error();
    break;
}
if(mysql_affected_rows() > 0) {
    print "Successful deletion<br>\n";
} else {
    print "deletion failed, due to unknown reasons. please try again<br>\n";
}

mysql_close($mysql_link);
} else if($HTTP_POST_VARS["action"] == "login") {
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "your id & pw are not valid. please try again<br>\n";
        exit;
    } else {
        /*connects to specific local database*/
        if( mysql_select_db($mysql_db) == FALSE) {
            print "could not select database: $mysql_db\n";
        }

        $email = $_POST['email'];
        $sql_show = "SELECT * FROM $mysql_table WHERE email=\"\$email\" AND origin=\"\$me\"";

        if( ($mysql_result = mysql_query($sql_show) ) == FALSE) {
            print "Query failed : " . mysql_error();
            break;
        }
        if(mysql_num_rows($mysql_result) > 0) {
            ?><form method="post" action="delete.php"><?php
            while( $line = mysql_fetch_array($mysql_result, MYSQL_ASSOC)) {
                $date_time = $line['date_pub'] . ";" . $line['time_pub'];
                ?><p><input type="radio" name="date_time"
                    value="<?php echo $date_time ?>">
                Topic: <a href="<?php echo $line['website']?>" target="_blank">

```

```

                <?php echo $line['topic']?></a><br>
Author: <a href="mailto:<?php echo $line['email']?>">
        <?php echo $line['author']?></a><br>
Affiliate Station: <?php echo $line['affiliate']?><br>
Published: <?php echo $line['date_pub']?><br>
Last Modified: <?php echo $line['date_mod']?><br>
Genre: <?php echo $line['genre']?><br>
Label: <?php echo $line['label']?> ...
        <?php echo $line['contact']?><br>
Review: <?php echo $line['review']?></p>
<?php
    }
    ?>
    <input type="hidden" name="id" value="<?php echo $_POST['id'] ?>">
    <input type="hidden" name="pw" value="<?php echo $_POST['pw'] ?>">
    <input type="hidden" name="email" value="<?php echo $_POST['email'] ?>">
    <input type="hidden" name="action" value="select_review">
    <input type="submit" name="submit" value="delete">
</form>
<?php
    }
    mysql_free_result($mysql_result);
    mysql_close($mysql_link);
}
} else {
print "you have chosen to delete a review from the local database. (NOTE: this will not
affect the rest of the ring.  to \"flush\" something from the entire ring, please
select \"flush review\" from the main menu). please enter login info...<br>\n";
?>
<form method="post" action="delete.php">
    DB id: <input type="Text" name="id"><br>
    password: <input type="password" name="pw"><br>
    email: <input type="text" name="email"><br>
    <input type="hidden" name="action" value="login">
    <input type="Submit" name="submit1" value="submit info">
</form>
<?php
}
include "footer.html";
?>

```

WWW SCRIPTS - FLUSH.PHP:

```
<?php
include "header.html";
include "funcs.php";
//error_reporting (E_ALL);
set_time_limit(0); /*sets time limit for execution time*/

if( ($config = file("config.dat")) == FALSE) {
    print "couldn't open file: config.dat\n";
    exit(1);
}
$prev_neighbor = trim($config[0]);
list($me, $pgp_id) = explode("::", trim($config[1]) );
$next_neighbor = trim($config[2]);
list($mysql_id, $mysql_pw) = explode("::", trim($config[3]) );
$webdir = trim($config[4]);

/*global config stuff*/
$port_q = 1108;

/*MySQL Stuff...check local db first*/
$mysql_host = "localhost";
$mysql_db = "localradio";
$mysql_table = "reviews";
$mysql_delim = "::";

if($_POST["action"] == "select_review") {
    /*create connection to mysql*/
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "Could not connect : " . mysql_error();
        break;
    }

    /*connects to specific local database*/
    if( mysql_select_db($mysql_db) == FALSE) {
        print "could not select database: $mysql_db\n";
        break;
    }
}
```



```

list($date, $time) = explode(";", $_POST['date_time']);
$email = $_POST['email'];

$where = "WHERE email=\"\$email\" AND date_pub=\"\$date\" AND time_pub=\"\$time\" \" \" .
        "AND origin=\"\$me\"";
$sql_del = "DELETE FROM $mysql_table " . $where;
$sql_que = "SELECT known_copies FROM $mysql_table " . $where;

if( ($mysql_result = mysql_query($sql_que) ) == FALSE) {
    print "query failed: " . $mysql_error();
    exit;
}

$sql_del_bin = asc2bin($sql_del);
$packet = "f:." . $sql_del_bin . "\n";

$servers = mysql_fetch_array($mysql_result, MYSQL_ASSOC);
$server_arr = explode(";", $servers['known_copies']);

if(mysql_query($sql_del) == FALSE) {
    print "could not delete: " . mysql_error();
    exit;
}
if(mysql_affected_rows() > 0) {
    print "Successful deletion<br>\n";
} else {
    print "deletion failed, due to unknown reasons. please try again<br>\n";
}

foreach($server_arr as $server) {
    if(strlen($server) > 1) {
        my_socket_write($server, $port_q, $packet);
    }
}

mysql_free_result($mysql_result);
mysql_close($mysql_link);

} else if($HTTP_POST_VARS["action"] == "login") {
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "your id & pw are not valid. please try again<br>\n";
    }
}

```

```

        exit;
    } else {
        /*connects to specific local database*/
        if( mysql_select_db($mysql_db) == FALSE) {
            print "could not select database: $mysql_db\n";
        }

        $email = $_POST['email'];
        $sql_show = "SELECT * FROM $mysql_table WHERE email=\"\$email\" AND origin=\"\$me\"";

        if( ($mysql_result = mysql_query($sql_show) ) == FALSE) {
            print "Query failed : " . mysql_error();
            break;
        }
        if(mysql_num_rows($mysql_result) > 0) {
            ?><form method="post" action="flush.php"><?php
            while( $line = mysql_fetch_array($mysql_result, MYSQL_ASSOC)) {
                $date_time = $line['date_pub'] . ";" . $line['time_pub'];
                ?><p><input type="radio" name="date_time"
                    value="<?php echo $date_time ?>"
                    Topic: <a href="<?php echo $line['website']?>" target="_blank">
                        <?php echo $line['topic']?></a><br>
                    Author: <a href="mailto:<?php echo $line['email']?>"
                        <?php echo $line['author']?></a><br>
                    Affiliate Station: <?php echo $line['affiliate']?><br>
                    Published: <?php echo $line['date_pub']?><br>
                    Last Modified: <?php echo $line['date_mod']?><br>
                    Genre: <?php echo $line['genre']?><br>
                    Label: <?php echo $line['label']?> ...
                        <?php echo $line['contact']?><br>
                    Review: <?php echo $line['review']?></p>
                <?php
            }
            ?>
            <input type="hidden" name="id" value="<?php echo $_POST['id'] ?>"
            <input type="hidden" name="pw" value="<?php echo $_POST['pw'] ?>"
            <input type="hidden" name="email" value="<?php echo $_POST['email'] ?>"
            <input type="hidden" name="action" value="select_review">
            <input type="submit" name="submit" value="flush">
            </form>
            <?php

```

```
        }
        mysql_free_result($mysql_result);
        mysql_close($mysql_link);
    }
} else {
    print "you have chosen to flush a review from the entire ring. (NOTE: to delete something
        exclusively from the local database, please select \"delete\" from the main menu).
        please enter login info...<br>\n";
    ?>
    <form method="post" action="flush.php">
        DB id: <input type="Text" name="id"><br>
        password: <input type="password" name="pw"><br>
        email: <input type="text" name="email"><br>
        <input type="hidden" name="action" value="login">
        <input type="Submit" name="submit1" value="submit info">
    </form>
    <?php
}
include "footer.html";
?>
```

WWW SCRIPTS - UPDATE.PHP:

```
<?php
    include "funcs.php";
    include "header.html";

    //error_reporting (E_ALL);
    set_time_limit(0); /*sets time limit for execution time*/

    if( ($config = file("config.dat")) == FALSE) {
        print "couldn't open file: config.dat\n";
        exit(1);
    }
    $prev_neighbor = trim($config[0]);
    list($me, $pgp_id) = explode("::", trim($config[1]) );
    $next_neighbor = trim($config[2]);
    list($mysql_id, $mysql_pw) = explode("::", trim($config[3]) );
    $webdir = trim($config[4]);

    $port_q = 1108;

    /*MySQL Stuff*/
    $mysql_host = "localhost";
    $mysql_db = "localradio";
    $mysql_table = "reviews";
    $mysql_delim = "::";

    $dir = "/tmp/ring/client/";
    $del_cmd = "rm -f " . $dir . "*";
    system($del_cmd);

    $keyring = "/www/pgp/pubring.pkr";
    putenv("PUBRING=$keyring");

    if($_POST['action'] == "update_submit") {
        if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
            print "your id & pw are not valid. please try again<br>\n";
            exit;
        }
        /*connects to specific local database*/
```

```

if( mysql_select_db($mysql_db, $mysql_link) == FALSE) {
    print "could not select database: $mysql_db\n";
    exit;
}

$author = $_POST['author'];
$email = $_POST['email'];
$new_email = $_POST['new_email'];
$affiliate = $_POST['affiliate'];
$topic = $_POST['topic'];
$genre = $_POST['genre'];
$website = $_POST['website'];
$label = $_POST['label'];
$contact = $_POST['contact'];
$review = $_POST['review'];

$date_pub = $_POST['date_pub'];
$time_pub = $_POST['time_pub'];

$date_mod = date("Y-m-d");
$time_mod = date("G:i:s");

$sql_ud = "UPDATE $mysql_table SET review=\"$review\",date_mod=\"$date_mod\",
    time_mod=\"$time_mod\",author=\"$author\",email=\"$new_email\",
    affiliate=\"$affiliate\",topic=\"$topic\",genre=\"$genre\",website=\"$website\",
    label=\"$label\",contact=\"$contact\" WHERE email=\"$email\" AND d
    date_pub=\"$date_pub\" AND time_pub=\"$time_pub\" AND origin=\"$me\"";

if( mysql_query($sql_ud) == FALSE) {
    print "could not update: " . mysql_error();
    exit;
} else print "Successful update to local db<br>\n";

$sql_que = "SELECT known_copies FROM $mysql_table
    WHERE email=\"$email\" AND origin=\"$me\"
    AND date_pub=\"$date_pub\" AND time_pub=\"$time_pub\"";
if( ($mysql_result = mysql_query($sql_que) ) == FALSE) {
    print "query failed: " . mysql_error();
    break;
}
}

$servers = mysql_fetch_array($mysql_result, MYSQL_ASSOC);

```

```

$server_arr = explode(";", $servers['known_copies']);

$sql_ud_bin = asc2bin($sql_ud);
$packet = "u::" . $sql_ud_bin . "\n";

foreach($server_arr as $server) {
    if(strlen($server) > 1) {
        my_socket_write($server, $port_q, $packet);
    }
}

/*Free result set */
mysql_free_result($mysql_result);

/*closing connection*/
mysql_close($mysql_link);

?><a href="main.php">click here to return to main menu</a><?php

} else if($_POST['action'] == "select_review") {
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "your id & pw are not valid. please try again<br>\n";
        exit;
    }
    /*connects to specific local database*/
    if( mysql_select_db($mysql_db, $mysql_link) == FALSE) {
        print "could not select database: $mysql_db\n";
        exit;
    }
}

list($date_pub, $time_pub) = explode(";", $_POST['date_time']);

$email = $_POST['email'];

$sql_que = "SELECT * FROM $mysql_table WHERE email=\"\$email\" AND origin=\"\$me\"
            AND date_pub=\"\$date_pub\" AND time_pub=\"\$time_pub\"";
if( ($mysql_result = mysql_query($sql_que) ) == FALSE) {
    print "query failed: " . mysql_error();
    exit;
}

```

```

?><form method="post" action="update.php"><?php

$line = mysql_fetch_array($mysql_result, MYSQL_ASSOC);
?>
Author: <input type="text" name="author" value="<?php echo $line['author']?>"><br>
Email: <input type="text" name="new_email" value="<?php echo $line['email']?>"><br>
Affiliate Station: <input type="text" name="affiliate"
                    value="<?php echo $line['affiliate']?>"><br>
Published: <?php echo $line['date_pub']?><br>
Last Modified: <?php echo $line['date_mod']?><br>
Topic: <input type="text" name="topic" value="<?php echo $line['topic']?>"><br>
Genre: <input type="text" name="genre" value="<?php echo $line['genre']?>"><br>
Artist Website: <input type="text" name="website" value="<?php echo $line['website']?>"><br>
Label: <input type="text" name="label" value="<?php echo $line['label']?>"><br>
Label Contact: <input type="text" name="contact" value="<?php echo $line['contact']?>"><br>
Review: <br><textarea name="review" cols=60 rows=15 wrap="wrap">
        <?php echo $line['review']?></textarea><br>

<input type="hidden" name="id" value="<?php echo $_POST['id'] ?>">
<input type="hidden" name="pw" value="<?php echo $_POST['pw'] ?>">
<input type="hidden" name="email" value="<?php echo $email ?>">
<input type="hidden" name="date_pub" value="<?php echo $date_pub ?>">
<input type="hidden" name="time_pub" value="<?php echo $time_pub ?>">
<input type="hidden" name="action" value="update_submit">
<input type="submit" name="submit" value="submit">
</form>
<?php

mysql_close($mysql_link);

}else if($_POST['action'] == "login") {
    if( ($mysql_link = mysql_connect($mysql_host, $_POST['id'], $_POST['pw'])) == FALSE) {
        print "your id & pw are not valid. please try again<br>\n";
        exit;
    }
    /*connects to specific local database*/
    if( mysql_select_db($mysql_db, $mysql_link) == FALSE) {
        print "could not select database: $mysql_db\n";
        exit;
    }
    $email = $_POST['email'];

```

```

$sql_que = "SELECT * FROM $mysql_table WHERE email=\"$email\" AND origin=\"$me\"";
if( ($mysql_result = mysql_query($sql_que, $mysql_link) ) == FALSE) {
    print "query failed: " . mysql_error();
    exit;
}
if(mysql_num_rows($mysql_result) > 0) {
    ?><form method="post" action="update.php"><?php

    while( $line = mysql_fetch_array($mysql_result, MYSQL_ASSOC)) {
        $date_time = $line['date_pub'] . ";" . $line['time_pub'];
        ?><p><input type="radio" name="date_time" value="<?php echo $date_time ?>">
        Author: <?php echo $line['author']?><br>
        Email: <?php echo $line['email']?><br>
        Affiliate Station: <?php echo $line['affiliate']?><br>
        Published: <?php echo $line['date_pub']?><br>
        Last Modified: <?php echo $line['date_mod']?><br>
        Topic: <?php echo $line['topic']?><br>
        Genre: <?php echo $line['genre']?><br>
        Artist Website: <?php echo $line['website']?><br>
        Label: <?php echo $line['label']?><br>
        Label Contact: <?php echo $line['contact']?><br>
        Review: <?php echo $line['review']?></p>
        <?php
    }
    ?>
    <input type="hidden" name="id" value="<?php echo $_POST['id'] ?>">
    <input type="hidden" name="pw" value="<?php echo $_POST['pw'] ?>">
    <input type="hidden" name="email" value="<?php echo $email ?>">
    <input type="hidden" name="action" value="select_review">
    <input type="submit" name="submit" value="submit">
</form>
<?php

} else {
    print "nothing to update here\n";
    exit;
}
/*Free result set */
mysql_free_result($mysql_result);

mysql_close($mysql_link);

```



```
} else {
    ?>
    <form method="post" action="update.php">
        You have chosen to update a review. Please enter...<br>
        site id: <input type="text" name="id"><br>
        site password: <input type="password" name="pw"><br>
        Your email: <input type="text" name="email"><br>
        <input type="hidden" name="action" value="login">
        <input type="submit" name="submit" value="submit">
    </form>
    <?php
}
include "footer.html";
?>
```

WWW SCRIPTS - SHOWDB.PHP:

```
<?php
include "header.html";
//error_reporting (E_ALL);
set_time_limit(0); /*sets time limit for execution time*/

if( ($config = file("config.dat")) == FALSE) {
    print "couldn't open file: config.dat\n";
    exit(1);
}
$prev_neighbor = trim($config[0]);
list($me, $pgp_id) = explode("::", trim($config[1]) );
$next_neighbor = trim($config[2]);
list($mysql_id, $mysql_pw) = explode("::", trim($config[3]) );
$webdir = trim($config[4]);

/*global config stuff*/
$port_q = 1108;

/*MySQL Stuff...check local db first*/
$mysql_host = "localhost";
$mysql_db = "localradio";
$mysql_table = "reviews";
$mysql_delim = "::";

$sql_show = "SELECT * FROM $mysql_table ORDER BY topic";
/*create connection to mysql*/
if( ($mysql_link = mysql_connect($mysql_host, $mysql_id, $mysql_pw)) == FALSE) {
    print "Could not connect : " . mysql_error();
    exit(1);
}
/*connects to specific local database*/
if( mysql_select_db($mysql_db) == FALSE) {
    print "could not select database: $mysql_db\n";
}

if( ($mysql_result = mysql_query($sql_show) ) == FALSE) {
    print "Query failed : " . mysql_error();
    break;
}
```

```
}
if(mysql_num_rows($mysql_result) > 0) {
    while( $line = mysql_fetch_array($mysql_result, MYSQL_ASSOC)) {
        ?>
        Topic: <a href="<?php echo $line['website']?>" target="_blank">
            <?php echo $line['topic']?></a><br>
        Author: <a href="mailto:<?php echo $line['email']?>">
            <?php echo $line['author']?></a><br>
        Affiliate Station: <?php echo $line['affiliate']?><br>
        Published: <?php echo $line['date_pub']?><br>
        Last Modified: <?php echo $line['date_mod']?><br>
        Genre: <?php echo $line['genre']?><br>
        Label: <?php echo $line['label']?> ... <?php echo $line['contact']?><br>
        Review: <?php echo $line['review']?></p>
        <?php
    }
}
/*Free result set */
mysql_free_result($mysql_result);
/*closing connection*/
mysql_close($mysql_link);

include "footer.html";

?>
```

WWW SCRIPTS - FUNCS.PHP:

```
<?php
/*create socket and connect*/
function my_connect($my_recipient, $my_port) {
    if(($my_socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP)) == FALSE) {
        print socket_strerror(socket_last_error()) . "\n";
        return $my_socket;
    }
    if(socket_connect($my_socket, $my_recipient, $my_port) == FALSE) {
        print socket_strerror(socket_last_error()) . "\n";
        return $my_socket;
    }
    return $my_socket;
}

/*create socket, connect, write string, close*/
function my_socket_write($my_recipient, $my_port, $my_packet) {
    $my_socket = my_connect($my_recipient, $my_port);

    if( socket_write ($my_socket, $my_packet, strlen($my_packet) ) == FALSE ) {
        print socket_strerror( socket_last_error() ) . "\n";
        return FALSE;
    }

    socket_close($my_socket);
    return TRUE;
}

/*create, bind, and listen*/
function my_listen($my_address, $my_port) {
    /*how many queued connections we can handle at one time*/
    $num_queue = 5;

    /* Create a new socket: */
    if( ($my_socket = socket_create( AF_INET, SOCK_STREAM, SOL_TCP )) == FALSE ) {
        print socket_strerror( socket_last_error() ) . "\n";
        return FALSE;
    }
}
```

```

/* Bind the socket to an address and a port: */
if( socket_bind( $my_socket, $my_address, $my_port ) == FALSE ) {
    print socket_strerror( socket_last_error() ) . "\n";
    return FALSE;
}

/* Listen for incoming connections on $my_socket */
if( socket_listen( $my_socket, $num_queue ) == FALSE ) {
    print socket_strerror( socket_last_error() ) . "\n";
    return FALSE;
}

return $my_socket;
}

/*puts string in file*/
function my_file_write($file, $string, $show) {
    switch($show) {
        case "b": $method = 'w+b';
                break;

        default: $method = 'w+';
                break;
    }

    $file_handle = fopen($file, $method);
    fwrite($file_handle, $string);
    fclose($file_handle);

    return TRUE;
}

/*converts binary string to ascii string*/
function bin2asc($temp) {
    $data = "";
    $len = strlen($temp);
    for($i=0; $i < $len; $i += 8)
        $data .= chr(bindec(substr($temp,$i,8)));
    return $data;
}

```

```
/*converts ascii string to binary string*/  
function asc2bin($temp) {  
    $data = "";  
    $len = strlen($temp);  
    for($i=0; $i < $len; $i++)  
        $data .= sprintf("%08b",ord(substr($temp,$i,1)));  
    return $data;  
}
```

?>