CS195-7: Introduction to Cryptography September 17-19, 2002

Lectures 5 and 6

Instructor:Anna Lysyanskaya

(This handout also contains some of the material covered in Lectures 2,3,4 for which I did not make careful lecture notes.)

1 Recap

1.1 Notation

Let A denote an algorithm. Let $A(\cdot)$ denote the fact that A has one input. $A(\cdot,\cdot)$ denotes the fact that A has two inputs.

$A(x)$ is a well-defined quantity if A is deterministic.

$A(x)$ defines a probability distribution if A is probabalistic. More precisely: Let $A'(\cdot,\cdot)$ be the deterministic algorithm such that $A'(x, R)$ is equal to $A(x)$ with the random tape R. Then $A(x)$ is the distribution induced by picking R uniformly at random and running $A'(x, R)$.

If A is a probabilistic algorithm, $y \in A$ denotes that y can be obtained by running A for some choice of random bits for algorithm A.

$y \leftarrow A(x)$ corresponds to choosing y from the distribution induced by $A(x)$.

$x \leftarrow F$, for a set F, means that x was selected from F uniformly at random.

We can now introduce notation that represents a sequence of experiments:

EXAMPLE 1

$((x, y) \leftarrow A(3); z \leftarrow B(y))$ means that A was run on the input 3, producing values x and y for output. Next B was run with input y and produced output z.

EXAMPLE 2

Let $B(x)$ be a boolean function of variable x.

To denote "Probability that $B(y)$ is true, given that y is generated by A being run on x," we write $Pr[y \leftarrow A(x) : B(y)]$

EXAMPLE 3

Let us formalize a statement such as "factoring RSA moduli is hard." What does this mean? First, it means that someone chose n at random. Then this n is given to the adversary. Then the adversary tries to factor n and produces a candidate factor p'. We wish to limit the probability that p' is a non-trivial factor of n, $\forall pptA_k$.

First, let us repeat this in English:

$Pr[$ 1. n gets chosen in some way;

2. n is given to the adversary and the adversary ouputs a candidate factor p' of n

$: p' \neq 1$ and $p' \neq n] \leq$ negligible function

And now let us rewrite this using our notation:

Let $PRIMES_k$ denote prime numbers of length k.
$\forall ppt A_k$, $\exists$ negligeble function f such that.
$Pr[p \leftarrow PRIMES_k; q \leftarrow PRIMES_k; n = p \cdot q\ p' = A_k(n)$
$: p' \neq 1$ and $p' \neq n$ and $p'|n] \leq f(k)$

1.2 The RSA Trapdoor Permutation

To recall the notation such as $\mathbb{Z}_n^*$, $\phi(n)$, etc, consult Dana Angluin's notes.

Setup: On input 1^k, choose two k-bit primes p and q, uniformly at random. Let $n = pq$. Choose a value e such that $\gcd(e, \phi(n)) = 1$. Compute d such that $d = 1/e \bmod \phi(n)$. Output the public key (n, e), and the trapdoor (secret key) d.

Evaluation: $f_{n,e}\ :\ \mathbb{Z}_n^* \mapsto \mathbb{Z}_n^*$ is defined as follows: $f_{n,e}(x) = x^e \bmod n$.

Inverstion: $f_{n,e}^{-1}(y) = y^d \bmod n$, where d is the secret key.

2 The Goldwasser-Micali Cryptosystem

Key generation: An RSA modulus $n = pq$ is chosen, as well as an element $u \in \mathbb{Z}_n^*$ such that u is not a square mod$p$ and mod$q$.

Encryption: To encrypt a message m, it is first divided up into bits: $m = m_1 \circ \ldots \circ m_\ell$. To encrypt the bit m_i, choose a value $r_i \leftarrow \mathbb{Z}_n^*$, and output $c_i = r_i^2 u^{m_i}$. This is equivalent to outputting a random square mod n in case m_i is 0, and a random non-square in case m_i is 1. The final ciphertext is $c = \{c_1, \ldots, c_\ell\}$.

Decryption: In order to decrypt a ciphertext $\{c_1, \ldots, c_\ell\}$, for each c_i, determine whether or not it has a square root. If yes, then $m_i := 0$, otherwise, $m_i := 1$.

Several mysteries remain about this cryptosystem at this point. First, why can key generation be done efficiently? Is the decryption efficient? Is the decryption faithful, i.e., does it retrieve the original message? Finally, why is this cryptosystem secure? Let us address these questions.

3 Some Useful Number-Theoretic Facts

Definition of the Legendre Symbol:
Let p be a prime number.

$$\left(\frac{a}{p}\right) \doteq \begin{cases} 0: & \text{if } GCD(a,p) \neq 1 \\ 1: & \text{if } GCD(a,p) = 1 \text{ and } a \text{ is a square mod } p \\ -1: & \text{if } GCD(a,p) = 1 \text{ and } a \text{ is not a square mod } p \end{cases}$$

Definition of Jacobi Symbol for RSA Moduli:

$$\left(\frac{a}{n}\right) \doteq \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

The term "square" can be used interchangeably with the term "quadratic residue" throughout.

Fact 0 (Fermat's little theorem): If p is prime, and a is relatively prime to p, then $a^p \equiv a \bmod p$.

Fact 1: $\mathbb{Z}_p^*$ is cyclic, i.e. there exists an element $g \in \mathbb{Z}_p^*$ such that for all elements $a \in \mathbb{Z}_p^*$ there exists a unique number $0 \le u \le p-1$ such that $a \equiv g^u \bmod p$. Such g is called a *generator* mod$p$.

Fact 2 (The Chinese Remainder Theorem): Let $m_1, \ldots, m_k$ be pairwise relatively prime integers. That is, $gcd(m_i, m_j) = 1$ for all $1 \le i \le j \le k$. Let the values $a_i \in \mathbb{Z}_{m_i}$ be given for $1 \le i \le k$. Let $m = \prod m_i$. There exists a unique $a \in \mathbb{Z}_m$ such that $a \equiv a_i \bmod m_i$ for all i. Furthermore there is an $O(k\ell^2)$ time algorithm to compute a given $a_1, \ldots, a_k$, $m_1, \ldots, m_k$, where $\ell = \max(|m_i|)$.

We won't give a proof of Facts 0, 1 and 2; see, for example, Goldwasser-Bellare's lecture notes for reference.

Fact 3: If $p > 2$ is prime, and g is a generator mod$p$, then g cannot be a square mod$p$. That is, for all $h \in \mathbb{Z}_p^*$, it is not the case that $h^2 \equiv g \bmod p$.

Proof: Suppose $\exists h$ such that $h^2 = g$, then $h^{p-1} = 1(\bmod\ p)$ by Fermats Little Theorem, and so $(h^2)^{\frac{p-1}{2}} = 1(\text{mod } p)$. Consequently $g^{\frac{p-1}{2}} = 1(\text{mod } p)$ but this is a contradiction because g is a generator. (Exercise: Why can't it be that for some $1 \le u < p-1$, $g^u \equiv 1 \bmod n$ if g is a generator?)

Fact 4: Let $p > 2$ be a prime. Let g be a generator mod$p$. Let $a \equiv g^u \bmod p$ be given. a is a square if and only if u is even.

Proof: Assume u is even, then $a = g^{2v} = g^{v2} \bmod\ p$ and a is obviously a quadratic residue.

Assume a is a quadratic residue. Suppose that u is odd, $u = 2v + 1$. Since a is a quadratic residue, $g^{2v+1} = b^2 \bmod\ p$ for some b.

$$g = \frac{b^2}{g^{2v}} \bmod\ p$$

$$= (\frac{b}{g^v})^2 \bmod\ p$$

but then g would be a quadratic residue which contradicts Fact 3.

Fact 5: Let $p > 2$ be a prime. Exactly one half of the elements of $\mathbb{Z}_p^*$ are quadratic residues mod$p$.

Proof: This is simply a corollary to Fact 4.

Fact 6: Let $p > 2$ be a prime. Let a be a square mod$p$. Then a has exactly two square roots mod$p$.

Proof: First, let us show that a has at least two square roots: By Fact 4, $a \equiv g^{2v}$ for some generator g and some number $1 \le v \le (p-1)/2$. Consider g^v and $g^{(p-1)/2+v}$. Observe that they must be distinct. Also observe that they are both square roots of a. Now, we must show that no number has more than two square roots modulo p. For that, we do a counting argument: by Fact 5, there are $(p-1)/2$ quadratic residues. We just saw that each of them has at least 2 square roots. That brings us to $p-1$ elememts accounted for. There is no more room for any other element to have any other square root.

Fact 7: Let $p > 2$ be a prime. Then 1 is a square mod$p$ and its only square roots are 1 and $-1 \bmod p$.

Proof: Fact 7 is simply a corollary of Fact 6.

Fact 8: Let p be a prime. Then $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$.

Proof: Let g be a generator. Let $a \equiv g^u \bmod p$, $b \equiv g^v \bmod p$. By Fact 4, $\left(\frac{ab}{p}\right) = 1$ if and only if $u + v$ is even, otherwise it's -1. On the other hand, $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = 1$ also if and only if $u + v$ is even, otherwise it's -1. Fact 6 follows.

Fact 9: The Legendre Symbol is computed by the following algorithm:

Test $p|a$, if p does divide a output 0, otherwise output $a^{\frac{p-1}{2}} \bmod \ p$

Proof: Case 1: a is a square. By Fact 4 $\exists v$ such that $a = g^{2v} \bmod \ p$

$$a^{\frac{p-1}{2}} = (g^{2v})^{\frac{p-1}{2}} = g^{v(p-1)} = 1 \bmod \ p$$

Thus we have shown that if a is a square, the Legendre Symbol is computed correctly.

Case 2: a is not a square. Then by Fact 4, $a \equiv g^{2v+1} \bmod p$. Therefore, $a^{(p-1)/2} \equiv (g^v)^{p-1} g^{(p-1)/2} \bmod p$. By Fermat's little theorem, we know that $(g^v)^{p-1} \equiv 1$. What about $g^{(p-1)/2} \bmod p$? First, observe that it is a square root of 1, since $g^{p-1} \equiv 1$. Now, notice that since g is a generator, it cannot be equal to 1. 1 has only two square roots, and by Fact 7 they are 1 and -1. We've ruled out 1, so it must be -1.

Fact 10: a is a quadratic residue mod $n = p_1 p_2$ if and only if a_1 is a square modp_1 and a_2 is a square modp_2, where $p_1 > 2$ and $p_2 > 2$ are primes.

Proof: Let us first show the $\Rightarrow$ direction. Assume a is a quadratic residue. We need to show that a_1 and a_2 are also quadratic residues.

$\exists b$ such that $b^2 = a \bmod \ n$.

Let $b_1 = b \bmod \ p_1$ and $b2 = b \bmod \ p_2$.

$$b_1^2 = b^2 = a = a_1 \bmod \ p_1$$

. Therefore, a_1 is a square. Similarly for b_2, which proves that a_2 is a quadratic residue as well.

Assume that a_1 and a_2 are quadratic residues.

$\exists b_1, b_2$ such that $a_1 = b_1^2 \bmod \ p$ and $a_2 = b_2^2 \bmod \ q$.Then by the Chinese Remainder Theorem, $\exists b$ such that $b = b_1 \bmod \ p$ and $b = b_2 \bmod \ q$.

$$b^2 = a_1 \bmod \ p$$

$$b^2 = a_2 \bmod \ p$$

Therefore since a is unique, $b^2 = a \bmod \ n$ by the Chinese Remainder Theorem.

Fact 11: Let n be an RSA modulus (i.e., $n = p_1 p_2$, where $p_1 > 2$ and $p_2 > 2$ are primes). Exactly one quarter of the elements of $\mathbb{Z}_n^*$ are quadratic residues.

Proof: Easy to see by combining Fact 10 with Fact 5.

Fact 12: Let $n = pq$ be an RSA modulus. Exactly half of the elements of $\mathbb{Z}_n^*$ have Jacobi Symbol 1.

Proof: This is obtained by a counting argument. Consider $a \in \mathbb{Z}_p^*$, $b \in \mathbb{Z}_q^*$. By the Chinese Remainder Theorem, there is a unique element c that corresponds to them. Moreover, by definition, $\left(\frac{c}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{q}\right)$. There are four possibilities. Case 1: $\left(\frac{a}{p}\right) = 1$, $\left(\frac{b}{q}\right) = 1$, and so $\left(\frac{c}{n}\right) = 1$. Case 2: $\left(\frac{a}{p}\right) = 1$, $\left(\frac{b}{q}\right) = -1$, and so $\left(\frac{c}{n}\right) = -1$. Case 3: $\left(\frac{a}{p}\right) = -1$, $\left(\frac{b}{q}\right) = 1$, and so $\left(\frac{c}{n}\right) = -1$. Case 4: $\left(\frac{a}{p}\right) = -1$, $\left(\frac{b}{q}\right) = -1$, and so $\left(\frac{c}{n}\right) = 1$. By Fact 5, it is easy to see that exactly one quarter of c's correspond to each of these cases.

Fact 13: Let n be an RSA modulus. Let $J_1(n)$ denote the set of elements of $\mathbb{Z}_n^*$ mod n with Jacobi Symbol 1. Exactly half of the elements of $J_1(n)$ are squares.

Proof: This follows from the proof of Fact 12.

Fact 14: Let $n = pq$ be an RSA modulus. One can efficiently determine the quadratic character of any $a \in \mathbb{Z}_n^*$ given the factorization of n.

Proof: For example, this can be done by computing $\left(\frac{a}{p}\right)$ and $\left(\frac{a}{q}\right)$ and verifying that these are 1. By Fact 10 this gives the right answer, and by Fact 9 this can be done efficiently.

Fact 15: Let n be an RSA modulus. For any $a \in \mathbb{Z}_n^*$, one can efficiently compute $\left(\frac{a}{n}\right)$ *without the factorization of* n.

Proof: We will omit the proof, but you can find it in GB lecture notes.

Don't confuse the Jacobi symbol modulo n and the quadratic character modulo n. Fact 15 tells us that the Jacobi symbol can be computed even without the factorization of n, while the quadratic character modulo n is something that we hope can *only* be computed if the factorization of n is known.

4 Quadratic Residuocity Assumption

If n is an RSA modulus chosen as random, and a random $a \in \mathbb{N}_n^*$, with Jacobi symbol 1, then you can't tell whether a has a square root mod n or not.

Using our notation, the assumption is as follows:

Let $J_1(n)$ denote the selection of a random element of $\mathbb{Z}_n^*$ mod n with Jacobi Symbol 1.

Let $QC(a,n)$ denote the quadratic character of a modulo n. That is to say, $QC(a,n) =$ "square" if a is a square and $QC(a,n) =$ "non-square" otherwise.

$\forall ppt\{A_k\}\exists$ negligible function f such that $\forall k$:

$$Pr[p \leftarrow PRIMES_k; q \leftarrow PRIMES_k; a \leftarrow J_1(p*q); b \leftarrow A_k(n,a)$$
$$: b = QC(a,n)] \leq \frac{1}{2} + f(k)$$

5 Proof of Security of the GM Cryptosystem

Besides coming up with their cryptosystem, Goldwasser and Micali (GM) also (1) defined semantic security (as a relaxation of information-theoretic security; we discussed their definition in Lecture 2); (2) gave a definition that is easier to work with and proved that it is equivalent to semantic security; and (3) proved security of their cryptosystem under the definition that is easier to work with.

For now, we will focus on their definition of security for sending one bit in the public-key setting. We will state the definition, called GM-security. We will then prove the GM cryptosystem GM-secure under the quadratic residuocity assumption. In future lectures, we will extend the notion of GM-security to many bits, and then show that it is equivalent to semantic security.

5.1 Indistinguishability

The idea behind security based on computational assumption, is that no matter whether a ciphertext is drawn from the set of ciphertexts for the message "I love you," or ciphertexts for message "I hate you," it will look the same to any computationally bounded adversary. So there are two samplable sets S_0 and S_1, such that a random sample from S_0 is indistinguishable from a random sample from S_1, and yet $S_0 \cap S_1 = \emptyset$. More formally:

Definition 5.1. *Two efficiently samplable distributions S_0 and S_1 cannot be $\nu(k)$-distinguished by algorithm $\{A_k\}$ if*

$$\Pr[b \leftarrow \{0,1\}; x \leftarrow S_b(1^k); b' \leftarrow A_k : b = b'] \leq 1/2 + \nu(k)$$

Definition 5.2. *Two efficiently samplable distributions S_0 and S_1 are computationally indistinguishable if for all probabilistic polynomial-time algorithms $\{A_k\}$ there exists a negligible function $\nu(k)$ such that S_0 and S_1 cannot be $\nu(k)$-distinguished by $\{A_k\}$.*

Definition 5.3. *Two efficiently samplable distributions S_0 and S_1 are statistically indistinguishable if for all (not necessarily bounded in any way) algorithms $\{A_k\}$ there exists a negligible function $\nu(k)$ such that S_0 and S_1 cannot be $\nu(k)$-distinguished by $\{A_k\}$.*

Definition 5.4. *Two efficiently samplable distributions S_0 and S_1 are perfectly indistinguishable if they are indentical.*

We will see indistinguishability pop up again and again! Now it is popping up for the purposes of secure encryption.

5.2 GM-Security for One-Bit Messages

Informally, a public-key cryptosystem is secure for sending a one-bit message if the distribution C_0 of ciphertexts for 0 is computationally indistinguishable from the distribution C_1 of ciphertexts for 1, and yet $C_0 \cap C_1 = \emptyset$.

More formally: A public key cryptosystem (G, E, D) is GM-secure if

Key generation The key generation algorithm, denoted by G, is a probabilistic polynomial-time algorithm that takes as its sole input the security parameter k. However, since traditionally we say that an algorithm is polynomial-time if its running time is polynomial in the size of its input, G's input k will be encoded in unary, denoted 1^k. The output of G is the key pair, (PK, SK). Using the notation introduced above, we write:

$$(PK, SK) \leftarrow G(1^k)$$

We say that the secret key SK *corresponds* to the public key PK if (PK, SK) can be produced by running $G(1^k)$.

Encryption The encryption algorithm E is a probabilistic polynomial-time algorithm that takes as input a bit b and the public key PK and produces as output the ciphertext c:

$$c \leftarrow E(b, PK)$$

such that $E(0, PK) \cap E(1, PK) = \emptyset$, i.e., if c is a ciphertext for b it cannot also be a ciphertext for $\bar{b}$, no matter what the random bits were that were used for key generation and encryption.

Decryption The decryption algorithm D is a probabilistic polynomial-time algorithm that takes as input the ciphertext c and decides whether $c \in E(0, PK)$ or $c \in E(1, PK)$.

Security The distributions $E(0, PK)$ and $E(1, PK)$ are computationally indistinguishable. More precisely: for all probabilistic polynomial-time families of algorithms $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[(PK, SK) \leftarrow G(1^k); b \leftarrow \{0,1\}; c \leftarrow E(b, PK); b' \leftarrow A_k(PK, c) : b = b'] = 1/2 + \nu(k)$$

5.3 Quadratic Residuocity Assumption

Recall that no public-key cryptosystem exists if $P = NP$, and so security of a public-key cryptosystem can only be proved under an assumption about the complexity of computational problems. Goldwasser and Micali proved that their cryptosystem was secure under the quadratic residuocity assumption.

In a nutshell, the quadratic residuocity assumption is the assumption that quadratic residues (QR) modulo an RSA modulus are computationally indistinguishable from quadratic non-residues with Jacobi symbol 1. More precisely:

Let RSA_k denote the set of RSA moduli $n = pq$ where $|p| = |q| = k$.

Let $J_1(n)$ denote the set of elements of $\mathbb{Z}_n^* \bmod n$ with Jacobi Symbol 1.

Let $QC(u, n)$ denote the quadratic character of u modulo n. That is to say, $QC(u, n) =$ "square" if u is a square and $QC(u, n) =$ "non-square" otherwise.

Assumption 5.1 (Quadratic residuocity assumption). *For all probabilistic polynomial-time families of algorithms $\{A_k\}$, there exists a negligible function $\nu(k)$ such that*

$$\Pr[n \leftarrow RSA_k; a \leftarrow J_1(n); b' \leftarrow A_k(n, a) \;:\; b' = QC(a, n)] = 1/2 + \nu(k)$$

Let $QR(n)$ denote the set of quadratic residues modulo n. Let $QNR(n)$ denote the set of quadratic non-residues modulo n with Jacobi symbol 1. Note that $a \leftarrow (QR(n) \cup QNR(n))$ is equivalent to $a \leftarrow J_1(n)$.

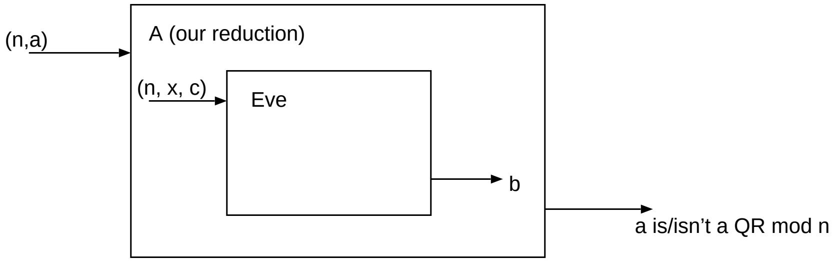
Recall (Fact 13) that exactly half of the elements of $J_1(n)$ are squares, so in fact $|QR(n)| = |QNR(n)|$. Then another, equivalent way of writing the selection of a is: $(b \leftarrow \{0,1\}; a_0 \leftarrow QR(n); a_1 \leftarrow QNR(n); a = a_b)$. That is to say, we first choose b, the quadratic character of a, and then choose a in accordance with b. In other words:

Assumption 5.2 (QR assumption, restated). *For all probabilistic polynomial-time families of adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that*

$$\Pr[n \leftarrow RSA_k; b \leftarrow \{0,1\}; a_0 \leftarrow QR(n); a_1 \leftarrow QNR(n); a = a_b; b' \leftarrow A_k(n,a) : b = b']$$

6 Proof of Security for One-Bit Messages

We want to show that if there exists an Eve who can efficiently figure out the encrypted bit, then the quadratic residuosity assumption is false.



So we need to construct an algorithm A that receives, as input, the values (n, a) and that uses Eve as a "black box." That is, it creates the inputs that Eve expects to receive, and interprets the output. In the event that Eve breaks the security of the encrypted bit, our algorithm A will break the QR assumption, i.e. it will, with probability non-negligibly better than 1/2, determine the quadratic character of a.

Assume, for contradiction, that there exists an adversary Eve such that there exists an inverse-polynomial $p(k)$ such that for infinitely many k,

$$\begin{aligned}\Pr[\,&(PK, SK) \leftarrow G(1^k);\\ &b \leftarrow \{0,1\};\\ &c \leftarrow E(PK, b);\\ &b' \leftarrow Eve(PK, c) : \quad b' = b] \geq 1/2 + p(k)\end{aligned}$$

Consider the following algorithm: on input (n, a), set $PK = (n, a)$. Then, choose b,r at random and let $c = r^2 a^b \bmod n$. Obtain $b' = \mathrm{Eve}(PK, c)$. If $b' = b$, output "Non-square" (or 0) Otherwise, output "Square" (or 1).

Claim 1: if a is a non-square, then the probability that $b' = b$ in this experiment is the same as the probability that Eve guesses the encrypted bit correctly. More precisely,

$$\begin{aligned}\Pr[\,&n \leftarrow RSA_k;\\ &a \leftarrow QR(n);\\ &b \leftarrow \{0,1\};\\ &r \leftarrow \mathbb{Z}_n^*;\\ &c \leftarrow r^2 a^b;\\ &b' \leftarrow Eve((n,a), c) : \quad b' = b] \geq 1/2 + p(k)\end{aligned}$$

Proof: The claim follows because in this case we are copying the GM cryptosystem set-up exactly. The first two steps comprise G, the next step is the selection of the message, and the next two steps are the encryption E. This is exactly what we have assumed that *Eve* can do. □

Claim 2: if a is a square, then the probability that $b' = b$ is exactly 1/2. More precisely:

```
$\forall PPTF\{Eve_k\}$
    $\Pr[\ n \leftarrow RSA_k;$
        $a \leftarrow QR(n);$
        $b \leftarrow \{0,1\};$
        $r \leftarrow \mathbb{Z}_n^*;$
        $c \leftarrow r^2 a^b;$
        $b' \leftarrow Eve_k((n,a),c): \quad b' = b] = 1/2$
```

Proof (intuition): Suppose a is a quadratic residue. Then, no matter what b is, c is distributed in the same way, i.e. c is always a random quadratic residue. Thus b is distributed independently on (n, a, c). Therefore, we can view Eve's output b' as fixed before b is even decided. Therefore, the probability that b will come out equal to b' is exactly 1/2.

Proof (formal): Suppose a is a quadratic residue and n as given. Then, $E((n, a), 0)$ and $E((n, a), 1)$ have distributions identical to that of a randomly selected quadratic residue. Consider a quadratic residue r. Observe first that $|QR| = \varphi(n)/4 = (p-1)(q-1)/4$. Thus, for a quadratic residue chosen uniformly at random from the space of quadratic residues modulo n, probability of selecting r is $|QR|^{-1}$, or $4/\varphi(n)$.

Now, let us consider $p_1 = \Pr[r' \leftarrow E((n,a),1) : r' = r]$. This means we take a random x^2 and transform it by multiplication by a. Since a is a quadratic residue (by assumption), the product is also a quadratic residue. We show that this is a bijective map.

one-to-one We need a^{-1} to exist; but since this is a subset of $\mathbb{Z}_n^*$, it does.

onto We must show that given y^2, there is x^2 such that $y^2 = ax^2$. Let $\alpha^2 = a$. Then $(y/\alpha)^2 = x^2$.

Thus, p_1 is the same as the probability of selecting any quadratic residue, or $4/\varphi(n)$.

It is clear that $\Pr[r' \leftarrow E((n,a),0) : r' = r]$ is also $4/\varphi(n)$ since encrypting a zero involves exactly selecting a random number and squaring it.

Since these three are identically distributed then, we can rewrite the Claim 2 as:

```
$\forall PPTF\{Eve_k\}$
    $\Pr[\ n \leftarrow RSA_k;$
        $a \leftarrow QR(n);$
        $c \leftarrow QR(n);$
        $b' \leftarrow Eve_k((n,a),c);$
        $b \leftarrow \{0,1\}: \quad b' = b] = 1/2$
```

That is, since encryption produces essentially a random quadratic residue (if a is a QR), Eve's output is essentially fixed: it does not matter what b is chosen. Since we are randomly selecting the plaintext, Eve's output will match ours half the time. □

To complete the proof of the reduction, note that:

$$\begin{aligned}\Pr[A \text{ is correct}] &= \Pr[A \text{ is correct } | a \in QR(n)] \Pr[a \in QR(n)] + \\ &\quad \Pr[A \text{ is correct } | a \in QNR(n)] \Pr[a \in QNR(n)] \\ &= \frac{1}{2} \cdot \frac{1}{2} + \left(\frac{1}{2} + p(k)\right) \frac{1}{2} \\ &= \frac{1}{2} + \frac{p(k)}{2}\end{aligned}$$

This is a basic fact from probability where we condition on all the possible cases for a. And so if Eve's advantage over 1/2, $p(k)$, is non-negligible, then our algorithm A will also have a non-negligible advantage $p(k)/2$.

More formally, the probability that A is correct can be written:

$$\begin{aligned}\Pr[\ & n \leftarrow RSA_k; \\ & d \leftarrow \{0,1\}; \\ & a_0 \leftarrow QR(n); a_1 \leftarrow QNR(n); a \leftarrow a_d; \\ & d' \leftarrow A_k(a,n) : \quad d' = d]\end{aligned}$$

Conditioning on whether a is a QR, this can be rewritten as:

$$\begin{aligned} & \Pr[\ n \leftarrow RSA_k; a \leftarrow QR(n); d' \leftarrow A_k(a,n) \quad : d' = 0] \times \Pr[d \leftarrow \{0,1\} : d = 0] \\ + \ & \Pr[\ n \leftarrow RSA_k; a \leftarrow QNR(n); d' \leftarrow A_k(a,n) : d' = 1] \times \Pr[d \leftarrow \{0,1\} : d = 1]\end{aligned}$$

But, since we know that A_k really makes use of Eve_k, this is exactly equal to:

$$\begin{aligned} & \Pr[\ n \leftarrow RSA_k; a \leftarrow QR(n); b \leftarrow \{0,1\}; \\ & \quad r \leftarrow \mathbb{Z}_n^*; c \leftarrow r^2 a^b; b' = Eve_k((n,a),c) \quad : b' = b] \times \Pr[d \leftarrow \{0,1\} : d = 0] \\ + \ & \Pr[\ n \leftarrow RSA_k; a \leftarrow QNR(n); b \leftarrow \{0,1\}; \\ & \quad r \leftarrow \mathbb{Z}_n^*; c \leftarrow r^2 a^b; b' = Eve_k((n,a),c) \quad : b' = b] \times \Pr[d \leftarrow \{0,1\} : d = 1]\end{aligned}$$

Fortunately, the claims above allow us to substitute in values. Claim 2 allows us to substitute exactly 1/2 × 1/2 for the first addend and Claim 1 says that the second addend is greater than $(1/2 + p(k)) \times 1/2$. If we let p be the above probability, then:

$$\begin{aligned} p &\geq \left(\frac{1}{2}\right) \frac{1}{2} + \left(\frac{1}{2} + p(k)\right) \frac{1}{2} \\ &\geq \frac{1}{2} + \frac{p(k)}{2}\end{aligned}$$