

1. **Classical cryptography.** Consider two classical cryptosystems described below. One has perfect security and one doesn't. In each case, Alice is attempting to send Bob a $64n$ bit message composed of n 64-bit blocks $(m_1, m_2, \dots, m_n) \in (\{0, 1\}^{64})^n$. Before the session begins, Alice and Bob agree to a certain key of the form $(k_1, k_2, \dots, k_n) \in (\{0, 1\}^{64})^n$ and a certain encryption function E so that the ciphertext blocks that Alice transmits are $c_i = E(m_i, k_i)$ for $i = 1$ to n .

- **FIRST CRYPTOSYSTEM:** A fixed exponent e is chosen by Alice and Bob (treated below). Each message block m_i and key block k_i is viewed as an element of $\mathbb{Z}_{2^{64}}$. Encryption is defined by $E(m_i, k_i) = (m_i^e + k_i) \bmod 2^{64}$.
- **SECOND CRYPTOSYSTEM:** E is just the DES encryption function.

(a) (5 points) How should e be chosen so that the first cryptosystem is valid? Give as specific a criterion as you can come up with.

Solution: We should choose $e=1$. We can not choose $e=0$ because every message will go to the same value and thus we will not have a valid cryptosystem. If $e \neq 1$ for every value of e we get that $E((2^{64})^e, k_i) = k_i$, but it is also the case that $E(0^e, k_i) = k_i$, thus we do not have a valid cryptosystem.

(b) (5 points) Give formulas for the decryption functions D for each cryptosystem.

Solution: Since $e = 1$ we just do $D(c_i, k_i) = c_i - k_i$.

(c) (8 points) Which cryptosystem is perfect? Prove your assertion.

Solution: The first crypto system is perfect, because $|K| = |C| = |M|$ and every key is used with equal probability.

(d) (7 points) Which cryptosystem is imperfect? Prove your assertion.

Solution: The second cryptosystem is imperfect, because we have $|K| < |M|$ and we stated in class that if the key space is less than the message space a cryptosystem can not be perfect.

2. **RSA exponents.** Let $(n = pq, e)$ be an RSA public key. As usual, p and q are different odd primes and e is relatively prime to $(p - 1)(q - 1)$. A **weak RSA message** is an element $m \in \mathbb{Z}_n$ with the property that it encrypts to itself, i.e. $e_k(m) = m$. (Here we are assuming for that the message space is \mathbb{Z}_n rather than \mathbb{Z}_n^* .)

(a) (5 points) How many weak messages are there for $(n, e) = (143, 7)$? Justify. **Solution:** 21. Justification: just plug into the next formula.

(b) (10 points) Characterize the weak messages for general RSA keys (n, e) . In particular, you should find a formula for the number of weak messages as a function of p, q and e . **Solution:** Let $g_p = \gcd(e - 1, p - 1)$ and $g_q = \gcd(e - 1, q - 1)$. Let x_p be a primitive element in \mathbb{Z}_p^* and x_q be a primitive element in \mathbb{Z}_q^* .

Weak messages =

$$\text{CRT-PATCH} \left(\left(\{0\} \cup x_p^{\frac{p-1}{g_p}} \cdot \mathbb{Z}_{g_p} \right) \times \left(\{0\} \cup x_q^{\frac{p-1}{g_q}} \cdot \mathbb{Z}_{g_q} \right) \right)$$

Cardinality = $(1 + g_p)(1 + g_q)$

- (c) (10 points) We say that an exponent e is ideal for n if it has a minimal number of weak messages. Give an example of an ideal exponent for $n = 143$. What is the global minimum on the number of weak messages that an ideal exponent can have? Prove your claims. **Solution:** Global minimum = 9, because if e is relatively prime to $\phi(n)$, $e - 1$ must be even so must have gcd at least 2 with both $p - 1$ and $q - 1$ so must have a at least 9 fixed points $\text{CRT-PATCH}(\{0, 1, -1\} \times \{0, 1, -1\})$. The global minimum is achieved for $(n, e) = (143, 23)$ since $\text{gcd}(22, 10) = \text{gcd}(22, 12) = 2$

3. **Computational security of PKE.** Consider the following public key encryption scheme. Given the security parameter k :

- secret keys are chosen uniformly at random from the set

$$\{(n, a, d) \mid n \in P_k \cdot P_k, \sqrt{n} \notin \mathbb{Z}, a \in \mathbb{Z}_n^*, d \in \mathbb{Z}_{\phi(n)}^*\}$$

where $P_k = \{p \mid p \text{ is a prime s.t. } |p| = k\}$ and the notation $S \cdot T$ denotes all products of the form $s \cdot t$ with $s \in S$ and $t \in T$.

- the public key corresponding to the secret key (n, a, d) is the tuple (n, b, e) where b is the inverse of $a \pmod n$ and e is the inverse of $d \pmod{\phi(n)}$
 - message space is $M = \mathbb{Z}_n^*$
 - for public key $PK = (n, b, e)$, encryption is achieved via the function $E(m, PK) = b \cdot m^e \pmod n$
- (a) (5 points) What should the decryption function $D(c, SK)$ be? Here SK is the secret key corresponding to the public key PK and the c is the ciphertext $c = E(m, PK)$. **Solution:** $D(c, SK) = (ac)^d$ We know $c = bm^e$ thus $(abm)^{de} = 1m^{de} = m$.
- (b) (5 points) Show that there is a PPT key generation algorithm that can generate (SK, PK) pairs with probability negligibly close to uniform.

Solution: -Generate two large numbers p and q
 -test if p and q are prime using MR primality testing if one is not regenerate it again and retest it
 -generate a random number b less than $n=pq$ such that $\text{gcd}(n, b) = 1$
 -use extended Euclidean algorithm to find $b^{-1} = a$
 -generate a random number d less than $\phi(n) = (p-1)(q-1)$ such that $\text{gcd}(\phi(n), d) =$

1.

-use extended Euclidean algorithm to find $d^{-1} = e$

(c) (10 points) Show that the encryption scheme is not computationally secure.

Solution: This encryption is not secure because it is a PKE scheme and it is not randomized, and thus can not be secure. If we give two messages x, y to an encoder that gives us back the encryption of either x or y , since the encryption is deterministic we just encrypt x and y ourselves and see which one matches the one from the encoder, thus we do not have multiple message indistinguishability.

(d) (5 points) Suggest a method for fixing the security hole in the scheme. Argue informally why the fix results in a computationally secure PKE scheme.

Solution: We can add randomness as follows: $E(m, PK) = b(m+r)^e$ then we send r with our encryption.

4. **Zero Knowledge Identification Protocol.** Consider the following protocol that honest Peggy and Vera have agreed to follow so that Peggy can convince Vera of her identity. Before the protocol begins, a trusted third party Trevor gives Peggy and Vera a very large number n which is a product of two distinct equal sized primes. Neither Peggy nor Vera know the factorization of n . Peggy chooses t random numbers (x_1, x_2, \dots, x_t) and squares them obtaining $y_i = x_i^2 \pmod n$ for $i = 1$ to t . Peggy publishes (y_1, y_2, \dots, y_t) but keeps (x_1, x_2, \dots, x_t) secret.

Whenever Peggy needs to authenticate herself to Vera, Peggy and Vera act according to the following protocol:

Repeat the following k times:

1. Peggy chooses $r \in \mathbb{Z}_n^*$ uniformly at random and sends $s = r^2 \pmod n$ to Vera.
2. Vera chooses a bit sequence $(b_1, b_2, \dots, b_t) \in \{0, 1\}^t$ uniformly at random and sends (b_1, b_2, \dots, b_t) to Peggy.
3. Peggy computes and sends $u = r \cdot x_1^{b_1} \cdot x_2^{b_2} \cdot \dots \cdot x_t^{b_t} \pmod n$ to Vera.
4. Vera computes a special boolean function B . If B is false, Vera sends "ABORT" and the protocol exists with Peggy not authenticated. On the other hand, if B is true, Vera sends the empty string " ε " and the protocol continues.

If the steps above were fully run k times with no ABORT signal, Vera ACCEPTS Peggy's authenticity.

(a) (5 points) Give a definition for Vera's boolean function B so that the above becomes a Zero Knowledge Interactive Protocol.

Solution: $B(s, (b_1, b_2, \dots, b_t), u) = "sy_1^{b_1}y_2^{b_2} \dots y_t^{b_t} \pmod n == u^2 \pmod n"$ where " $==$ " is the boolean function that test for equality.

(b) (2 points) Prove that the resulting Interactive Protocol is complete.

Solution: If P and V are honest, then $sy_1^{b_1}y_2^{b_2}\cdots y_t^{b_t} \equiv_n r^2(x_1^2)^{b_1}(x_2^2)^{b_2}\cdots(x_t^2)^{b_t} \equiv_n (r(x_1)^{b_1}(x_2)^{b_2}\cdots(x_t)^{b_t})^2 \equiv_n u^2$ so B always evaluates to true (probability = 1).

(c) (8 points) Prove that the resulting Interactive Protocol is sound. (What values of t, k should be chosen?)

Solution: Consider Pernicia (P^*) interacting with Vera for one run through of the protocol ($k=1$).

CLAIM: If Pernicia's probability of fooling Vera is significantly greater than $\frac{1}{2^t}$, then Pernicia can be employed as a black box in an algorithm that can factor n with non-negligible probability.

Proof: (Thanks you Pinxing Ye for noting a mistake in the previous solution). Here's an outline of the proof:

1. Show that if Pernicia can take the public information n, y_i and Vera's $\beta = (b_1, \dots, b_t)$ and output s, u_β with probability significantly better than $1/2$ then Pernicia can factor one of the y_i with significant probability (call it $p \gg \varepsilon$).
2. To turn Pernicia into a factoring algorithm, show that she can find square roots of arbitrary quadratic residues, by taking a desired quadratic residue q , embedding it inside the random public information (y_1, \dots, y_t) and then running Pernicia so with probability p/t she recovers the square root of y .
3. By repeating k times, we can make Pernicia's success probability become negligibly small.

Further details about (1) above (the only type of argument we haven't seen before): Consider all possible vectors β supplied by Vera. Since Pernicia succeeds much more than $1/2$ of the time, there must be two choices β, β' which agree on every coordinate except one (this follows from the pigeon-hole principle). Use the shorthand $y^\beta = y_1^{b_1} \cdot y_2^{b_2} \cdots y_t^{b_t}$. Consider the vector $\beta - \beta'$ which WLOG is 1 at exactly one coordinate say coordinate i_* . Notice that

$$\left(\frac{u_\beta}{u_{\beta'}} \right)^2 \equiv_n y_{i_*}$$

Thus Pernicia can find a square root of y_{i_*} —a random number— with high enough probability to complete the proof.

In general, the probability of Pernicia fooling Vera cannot be significantly greater than $\frac{1}{2^{kt}}$ and to obtain soundness just choose this value to be significantly less than $\frac{1}{3}$.

(d) (10 points) Prove that the resulting Interactive Protocol is Zero Knowledge. **Solution:** We need to show that a simulator Simon can generate valid transcripts with probability almost equal to the Peggy-Vera transcripts. We do the case $k = 1$. Bigger k are similar. The simulator runs as follows

4261 Final (continued)

1. Simon hopes Vera will choose random β , generates a random r and transmits $s = \frac{r^2}{y^\beta} \bmod n$ (using same notation as in part (c))
2. Vera gives β'
3. If Simon guessed wrong, he goes back to step (1)
4. Else, Simon transmits $u = r$.

Notice that we have

$$sy^\beta \equiv_n \frac{r^2}{y^\beta} y^\beta \equiv r^2$$

which means that the predicate $B(s, \beta, u)$ evaluates to true so the resulting transcript is valid. Furthermore, all transcripts occur with equal likelihood because all β, s are equally likely, and u is parameterized by these quantities.

The only remaining issue is that the simulator seemingly fails too often (if we consider going back to step 1 a failure). In fact, Simon needs to rewind to step 1 with probability $1 - \frac{1}{2^t}$. However, t only need be chosen so that the scheme is sound so that though $1 - \frac{1}{2^t}$ is close to 1, it is not negligibly close to 1. Therefore, by letting Simon rewind polynomially many times, Simon is able to simulate any transcript with probability negligibly close to the probability of a real transcript and the protocol is zero-knowledge.