

## 1

a. Suppose we are working in group  $Z_p$  with generator  $g$ , and we are given a particular instance of the DL problem,  $h = g^i$ . Then we can transform it into a random instance of DL as follows: we choose random  $y \leftarrow Z_{p-1}$ , and compute  $h' = h * g^y$ . If we could solve the DL problem on this random instance, and obtain  $z$  such that  $h' = g^z$ , then we could observe that  $g^z = hg^y$ , so  $h = g^{z-y}$ , which implies that  $z-y$  is a discrete log of  $h$ . Thus, we could use a solution to our randomly reduced problem to solve our original problem, which means that DL is randomly self-reducible.

Now, we assume there exists some  $\mathcal{A}$  that can solve random instances of the DL problem with nonnegligible probability. Then our algorithm  $\mathcal{B}$  on input  $x$ , picks a random  $y$  in  $\mathbb{Z}_p$  and computes  $g^y$ . He then gives  $\mathcal{A}$  the value  $z = x \cdot g^y$ . Note that this is necessary, because the  $x$  that  $\mathcal{B}$  is given is  $x = g^i$  where  $i \in \mathbb{Z}_{p-1}^*$ , but multiplying by the  $g^y$  will produce a completely random value  $z$ , so that  $\mathcal{A}$  cannot be purposefully unhelpful.  $\mathcal{A}$  will return some value  $l$  such that  $z \equiv g^l \pmod{p}$ . Since  $z = x \cdot g^y \equiv g^l$ ,  $\mathcal{B}$  computes  $(g^y)^{-1}$  and multiplies both sides to get  $x \equiv g^{l-y} \pmod{p}$ . Since  $\mathcal{B}$  knows both  $l$  and  $y$ , this gives a value  $i' = l - y$  such that  $x \equiv g^{i'} \pmod{p}$ . Therefore, we have showed that if there exists some  $\mathcal{A}$  to break the DLP, there must exist some  $\mathcal{B}$  to break the modified assumption, or that if the modified assumption holds, the DLP is hard as well.

b. First, we assume that there exists some  $\mathcal{A}$  that can break the modified DLP with non-negligible probability  $\epsilon$ . Then, since it will only work for  $i \in \mathbb{Z}_{p-1}^*$ , we need to make sure that this group has a large enough size. Since  $p$  is  $k$ -bits,  $p-1$  will be  $k$ -bits as well. By the prime number theorem, there are  $\approx \frac{p-1}{\log(p-1)} > \frac{p-1}{k-1}$  prime numbers between 1 and  $p-1$ . Only very few of these numbers can be prime factors of  $p-1$ , so we can say that there should be at least  $\frac{p-1}{k-1}$  elements in this group  $\mathbb{Z}_{p-1}^*$ . Now,  $\mathcal{B}$  will just run whatever  $x$  he is given through  $\mathcal{A}$ . If  $x = g^i$  for  $i \in \mathbb{Z}_{p-1}^*$ ,  $\mathcal{A}$  will correctly return the discrete log. Otherwise, it will fail. There are  $p-1$  elements in  $\mathbb{Z}_p^*$ , so  $\mathcal{B}$  has probability  $\frac{\frac{p-1}{k-1}}{p-1} = \frac{1}{k-1}$  of getting an element that works. So  $\mathcal{B}$  will succeed with at least  $\frac{\epsilon}{k-1}$  probability, which is not negligible.

## 2

a. We say that  $H_{PK} : D_{PK} \rightarrow R_{PK}$  (where  $D_{PK}$  and  $R_{PK}$  just represent some domain/range corresponding to  $PK$ ) is a collision-resistant hash function if, for all probabilistic polynomial-time adversaries  $\mathcal{A}$  there exists some negligible function  $\nu$  such that

$$\Pr[PK \leftarrow \text{HKeyGen}(1^k) ; (x, x') \leftarrow \mathcal{A}(PK) : x \neq x', x, x' \in D_{PK}, H_{PK}(x) = H_{PK}(x')] = \nu(k).$$

b. First, we note that a collision means (for some  $n$  and  $y$ ) that we have an  $x \neq x'$  such that  $H_{(n,y)}(x) = y^x = y^{x'} = H_{(n,y)}(x')$ . So, we have  $x \neq x'$  such that  $y^x = y^{x'}$ , which means that  $y^{x-x'} \equiv 1 \pmod{n}$  and  $x - x' \equiv 0 \pmod{\phi(n)}$ . Therefore,  $x - x' = k\phi(n)$  for  $k \in \mathbb{Z}$ , and since we know that  $x \neq x'$  we can be sure this is a non-zero multiple. Now, to break RSA we need to find (for a given  $e$  which is relatively prime to  $\phi(n)$ ) a  $d$  satisfying  $ed \equiv 1 \pmod{\phi(n)}$ , or  $ed = 1 + l\phi(n)$  for  $l \in \mathbb{Z}$ . Normally, all we need to do is perform the extended Euclidean algorithm using  $e$  and  $\phi(n)$

to find such a value. But, all we have is  $k\phi(n)$ . Note that if  $e, k$  are relatively prime, then  $e, k\phi(n)$  will be relatively prime, so we can perform the extended Euclidean algorithm using  $e$  and  $k\phi(n)$  to get a  $d$  such that  $1 = ed + l'k\phi(n)$ . That means  $ed = 1 \pmod{\phi(n)}$ , so we can use  $d$  to invert RSA.

Thus, the only problem comes if  $e$  and  $k$  are not relatively prime. To solve this problem we divide by  $\gcd(e, k\phi(n))$ , which since  $\gcd(e, \phi(n)) = 1$  is just equal to  $\gcd(e, k)$ . Now, we perform the extended Euclidean algorithm using  $e$  and  $\frac{k\phi(n)}{\gcd(e, k\phi(n))}$  to get a  $d$  such that  $1 = ed + l \cdot \left(\frac{k\phi(n)}{\gcd(e, k\phi(n))}\right)$ . If we call this gcd value  $m$ , we can see this is written  $1 = ed + l \cdot \frac{k}{m}\phi(n)$ . By our argument above,  $m = \gcd(e, k)$ , which means that  $m \mid k$  and  $\frac{k}{m}$  is an integer. Therefore,  $\frac{lk}{m}$  is also an integer, and we have found a  $d$  such that  $ed = 1 - j\phi(n)$  for  $j \in \mathbb{Z}$ , so  $ed \equiv 1 \pmod{\phi(n)}$ .

### 3

a. We want to show that if factoring is hard, it is also hard to make the scheme target-message forgeable using a public-key only attack. We are, of course, going to prove this using the contrapositive: if there exists an adversary  $\mathcal{A}$  who can make the scheme target-message forgeable with a PK-only attack, we can construct a  $\mathcal{B}$  to factor  $n$  with non-negligible probability. So, we assume that we have some  $\mathcal{A}$  such that

$$\Pr[PK \leftarrow \text{Gen}(1^k); m \leftarrow \text{QR}_n \cup \text{QNR}_n; b \leftarrow \mathcal{A}(PK, m) : \text{Verify}(m, b) = \text{Yes}] = \epsilon$$

for some non-negligible  $\epsilon(k)$ . Our algorithm  $\mathcal{B}$  picks some random  $r$  in  $\mathbb{Z}_n^*$  and computes  $x \equiv r^2 \pmod{n}$ . It then gives  $x$  to  $\mathcal{A}$  (along with the PK, which in this case is just  $n$ ), and takes the  $b$  that  $\mathcal{A}$  returns. If  $b^2 \equiv x \pmod{n}$  but  $b \not\equiv \pm r \pmod{n}$ , we have seen that  $\mathcal{B}$  can factor  $n$  (see next paragraph for a reminder how). Otherwise,  $\mathcal{B}$  outputs “fail.”

If we have  $b^2 \equiv r^2 \pmod{n}$ , we know that  $b^2 - r^2 = kn$  for  $k \in \mathbb{Z}$ . Then  $(b - r)(b + r) = kn$ , which is the same as saying that  $n \mid (b - r)(b + r)$ . Because  $b \not\equiv r \pmod{n}$  we know that  $n \nmid b - r$ , and because  $b \not\equiv -r \pmod{n}$ , we know that  $n \nmid b + r$ . Then there must exist some non-trivial divisor of  $n$ , call it  $p$ , such that  $p \mid (b + r)$  and  $p \mid n$ . So  $p = \gcd(n, b + r)$ , and once we have this value we have  $q$  and  $n$  is completely factored. So our algorithm works to factor  $n$  because it is sufficient to find two square roots with this property.

The only problem with this approach is that  $\mathcal{A}$  might only work when given  $x \in \text{QNR}_n$ . We sometimes make the assumption that  $\text{QNR}_n \approx \text{QR}_n$ , but in this case we would like to prove our scheme secure based only on assumption that factoring is hard (which is generally considered to be a weaker assumption). Thus, we need to modify the reduction above so that it randomly passes  $\mathcal{A}$  either  $r^2$ , or  $-r^2$ . The rest of the reduction will proceed as described above.

b. For this attack, our adversary should pick the signature  $\sigma$  first. Then, they can simply compute  $\sigma^2 \pmod{n}$  to get something that is in the message space (in fact, specifically in  $\text{QR}_n$ ), and has  $\sigma$  as its corresponding signature.

c. Our adversary first picks a random  $x$  in  $\mathbb{Z}_n^*$ . He then computes  $m = x^2$  and gives this to the signer. We get back some square root  $\sigma$  and check to see if  $\sigma \neq \pm x$ . Since there are four square

roots of  $m$  modulo  $n$ , we have probability  $\frac{1}{2}$  of this condition being met. As we have seen before, once we have two elements  $\sigma$  and  $x$  such that  $\sigma^2 \equiv x^2 \pmod{n}$  but  $\sigma \not\equiv \pm x \pmod{n}$ , we can find a non-trivial divisor of  $n$  and therefore factor it completely.