Lecture 14b: 3/10/04

1 Practical schemes and the random oracle model

In practice, people use cryptographic hash functions instead of the fancy hard-core bit permutations we discussed in the first half of lecture. SHA is an example of such a hash function:

$$H : \{0,1\}^{\phi} \to \{0,1\}^{160}$$

SHA passes statistical tests for randomness, but there is no proof that it is secure.

1.1 Several hybrid encryption schemes

Public key encryption is slow, so people typically use it to exchange private keys (e.g. AES keys). Then they use symmetric key cryptography for the bulk of the data. For instance, an RSA key exchange might look like this:

$$E_{\text{pk}}(k) = \text{pad}(k)^3 \pmod n$$

(Here $\text{pad}(x)$ means x padded to 128 bits.) This scheme is used in practice, but it is inelegant. We won't consider it because it is hard to analyze, but people have done so. Instead, we will look at the following three schemes, presented in reverse chronological order:

(A) $E_{\text{pk}}(m) = (r^3 \bmod n, H(r) \oplus m)$
random value $r \in (\mathbf{Z}/n\mathbf{Z})^{\phi}$

(B) $E_{\text{pk}}(m) = (r^3 \bmod n, \text{lsb}_{128}(r) \oplus m)$
random value $r \in (\mathbf{Z}/n\mathbf{Z})^{\phi}$

(C) $E_{\text{pk}}(m) = m^3 \bmod n$

Scheme C is clearly broken. If $m < n$, an eavesdropper can learn m by taking the cube root of the ciphertext. Moreover, the encryption function is deterministic, so repeated messages will have identical ciphertexts. (B) is an improvement, but its security is not known. For (B) to be secure, it is necessary (but perhaps not sufficient) for *all* of the lowest 128 bits of r to be hard-core. Do we need other assumptions as well?

In A, H mixes things up, but now we need to prove that $H(r)$ gives us hardcore bits. However, some choices of r are terrible. For example, if we encounter two outputs from H that collide, an eavesdropper can XOR the two ciphertexts together to obtain $m_1 \oplus m_2$. To prove anything about this scheme, we need a new model that allows us to reason about the protocol independently of the properties of H.

1.2 A new notion for security

1.2.1 First attempt

If we could prove that there are no *generic* attacks for all hash functions $H : \{0,1\}^{\phi} \to \{0,1\}^{160}$, then we might say that our scheme is reasonable. But this is not satisfying because it asks a lot of our attacker. Consider, for instance, an attack that works for H such that high-order bit has probability 50% of being set. Intuitively, all "good" cryptographic hash functions fit in this category, so we should take such an attack seriously; however, the above model does not cover it.

1.2.2 Second attempt

We could try to prove that no attack that works for random H. This is a stronger notion, but it is also not satisfying because the attacker is forced to choose an attack in advance without knowledge of the hash.

1.2.3 Third attempt: The random oracle model

In the random oracle model, we must show that there is no attack A such that A^H works with high probability over choice of H. This captures the notion of all hash functions with most significant bit balanced and it gives the attacker knowledge of the hash function. Moreover, it doesn't depend on specific properties of the hash function (e.g. SHA).

In this model, everyone has access to the black box H as depicted in Figure 1, but nobody has knowledge of the internals of this random function. Formally, we define the advantage of an attacker A in RO-IND-CPA (**ind**istinguishability under **c**hosen **p**laintext **a**ttacks in the **r**andom **o**racle model) in the real-or-random sense as follows:

$$\mathrm{Adv}\, A = \Pr_{\substack{(\mathrm{pk},\mathrm{tk})\leftarrow \mathrm{Gen}() \\ H}}[A^{H,E_{\mathrm{pk}}} = 1] - \Pr_{\substack{(\mathrm{pk},\mathrm{tk})\leftarrow \mathrm{Gen}() \\ H}}[A^{H,E_{\mathrm{pk}}\circ \$} = 1]$$

We say that E is (t, q, ϵ)-RR-secure iff $\mathrm{Adv}\, A \le \epsilon$. Note that q refers to the number of queries the attacker is allowed to make to the encryption oracle E_{pk}. We do not distinguish steps of computation from hash computations. Our previous analyses of PRGs can be extended to this model easily by simply taking the probability over H as well.

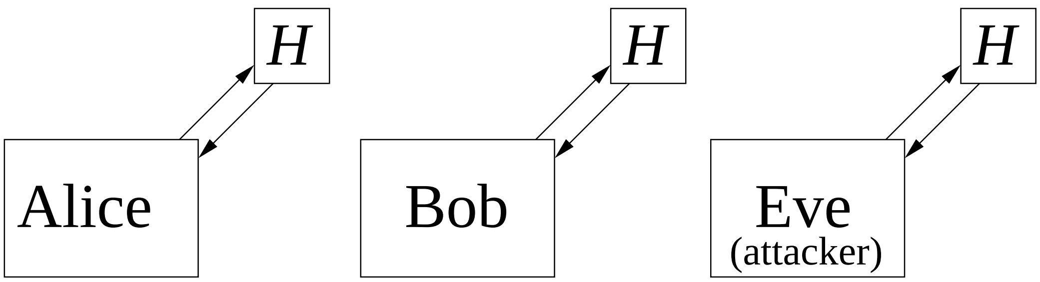

Figure 1: The setting for the random oracle model

1.3 Security of hybrid encryption

We can now apply the random oracle model to prove the security of scheme (A), introduced earlier.

THEOREM
If RSA is a (t, ϵ)-TOWP, then $E_{\text{pk}}(m) = (r^3 \bmod n, H(r) \oplus m)$ is $(t, 1, \epsilon)$-RR secure (RO-IND-CPA)

COROLLARY
If RSA is a (t, ϵ)-TOWP, E_{pk} is $(t, q, q\epsilon)$-secure by a hybrid argument.

The setup is illustrated in Figure 2. The difference between the real and random worlds is what the attacker gets from the E_{pk} box. In the real world, he gets $H(r) \oplus m$, and in the random world, he gets $H(r) \oplus u_0$, which is random.

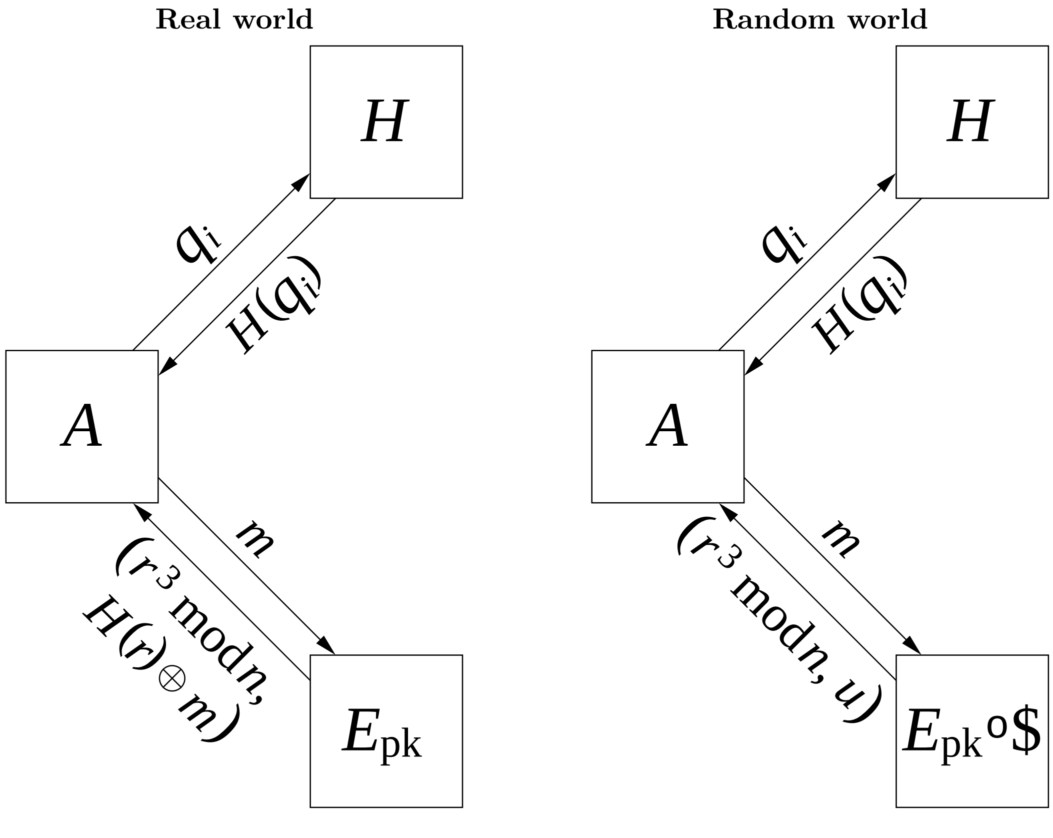

Figure 2: The setting for the random oracle model

Notice that if there is ever a collision in H, the attacker can tell that he is in the real world by finding r and sending it to H. However, finding r ought to be as hard as inverting RSA. The proof will show two things:

1. The probability of a collision is small.

2. Collisions are the only bad things that can happen.

To be more formal about the first condition, we define event COLL $\equiv (\exists \mathrm{i} : \mathrm{q_i} = \mathrm{r})$. We will prove the following.

1a. $Pr[\text{COLL}] \leq \epsilon$

1b. $Pr[A^{H,E_{\text{pk}}} = 1 | \text{no COLL}] = \mathrm{Pr}[\mathrm{A}^{\mathrm{H,E_{pk}\circ\$}} = 1 | \text{no COLL}]$

To show that a lack of collision is sufficient, we will show that if $H(r)$ has no collision, then it's close to random. The details will be covered in the next lecture.

References

[BR] M. Bellare and P. Rogaway. *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols.* ACM Conference on Computer and Communications Security, 1993.