University of Maryland
CMSC456 — Introduction to Cryptography
Professor Jonathan Katz

Lecture 38

1 Signature Schemes From Any One-Way Function

At the end of the last lecture, we saw how a secure signature scheme for arbitrarily many messages could be constructed from any 1-time signature scheme *which allows signing messages longer than the public key*. So far, however, the only construction of a 1-time signature scheme of this type was based on collision-resistant hash functions (CRHFs). We mentioned earlier in the course that the existence of CRHFs seems to be a strictly stronger assumption than the existence of one-way functions; in fact, it is believed to be impossible to construct CRHFs from an arbitrary one-way function. On the other hand, based on our experience with the Lamport 1-time signature scheme it may seem intuitive that one-way functions should be sufficient to construct a secure signature scheme or many messages. In fact, this can be done; we explore this matter here.

1.1 Universal One-Way Hash Functions

We first recall the definition of a collision-resistant hash function. We say H is (t, ϵ)-collision resistant if for all algorithms A running in time at most t we have:

$$\Pr[(x, y) \leftarrow A : H(x) = H(y) \wedge x \neq y] \leq \epsilon.$$

That is, the adversary A gets to output two values x and y, and succeeds if $x \neq y$ yet $H(x) = H(y)$.

A different notion is that of *universal one-way hash functions* (UOWHFs), which we introduce here. In this case, the adversary does not get to choose both x and y; instead, the adversary is given a random x and must find a different y such that $H(x) = H(y)$. More formally[1], let H be a keyed hash function so that $H_s : \{0,1\}^m \to \{0,1\}^n$ for all $s \in \{0,1\}^k$. We say this function is (t, ϵ)-universal one-way if for all algorithms A running in time t we have:

$$\Pr[x \leftarrow \{0,1\}^m; s \leftarrow \{0,1\}^k; y \leftarrow A(s, x) : H_s(x) = H_s(y) \wedge x \neq y] \leq \epsilon.$$

(Note: this definition is different than the one given in class since here we consider a *keyed* hash function.) Note that here both s and x are fixed in advance; the adversary only has control over y. This makes the adversary's job harder, meaning that UOWHFs are weaker than CRHFs. It should be immediately clear, for example, that any collision-resistant hash function is also universal one-way.

The following theorem was proved by Rompel [2], building on the work of Naor and Yung [1].

[1] The reader should be aware that, for simplicity, I have used a weaker definition of UOWHFs than appears in the literature. However, any hash function satisfying the present definition can be immediately converted to one satisfying the stronger definition.

Theorem 1 *Universal one-way hash functions can be constructed from any one-way function.*

We now show that UOWHFs are sufficient to construct secure signature schemes. Recall the tree-based signature scheme from the previous lecture, and remember why CRHFs were needed: because at node w of the tree we sign $PK_{w0}|PK_{w1}$ using PK_w. To do this (recalling the way we sign messages longer than the public key), we will first compute $h = H(PK_{w0}|PK_{w1})$ and then sign h using the regular Lamport scheme. Clearly, if H is collision resistant then this is secure.

We may note, however, that collision resistance is stronger that what we actually need *for this application*. The reason is that *the adversary doesn't choose* $PK_{w0}|PK_{w1}$; instead, this is a "random" pair of public keys chosen by the signer. Thus, we don't require that the adversary cannot find *any* collision in H; all we need is that the adversary cannot find a (different) $PK'_{w0}|PK'_{w1}$ such that $H(PK'_{w0}|PK'_{w1}) = H(PK_{w0}|PK_{w1})$. Since $PK_{w0}|PK_{w1}$ is chosen independent of the adversary, having H be a UOWHF[2] is enough!

Thus, the previous theorem as well as our results from the last lecture show that:

Theorem 2 *A secure signature scheme for arbitrarily-many messages can be constructed from any one-way function.*

We emphasize the amazing nature of this result: as first glance, it seems that some sort of "trapdoor" is necessary in order to sign. This intuition, however, is wrong: one-way functions (which have no "trapdoor") are sufficient to construct digital signature schemes. (This is in contrast to public-key encryption schemes, which are believed to be impossible to construct from one-way functions alone.)

2 The Random Oracle Model

The preceding section ends our discussion of schemes which can be rigorously proven secure based on well-defined, standard cryptographic assumptions. But where do we go from here? The reader will notice that — with the possible exception of the El Gamal encryption scheme — *all constructions of public-key encryption and digital signature schemes that we have given are inefficient*. In practice, the construction we have given are simply too inefficient to be used. What, then, is done? Do we simply use efficient schemes that have no argument in favor of their security?

The answer is (in general) **no**. When possible, we never want to use schemes which have no argument in favor of their security. By using a scheme which "seems hard to break" we leave ourselves open to an adversary who is more clever than us, and who can break the scheme. It is always better to have some justification for believing a scheme is secure.

Unfortunately, in general we simply do not know how to construct efficient schemes which are provably secure based on standard cryptographic assumptions. And yes, something must be done if we want to have efficient schemes. What is done is to introduce

[2]In fact, the careful reader will notice that having H satisfy Definition 1 is not quite enough for this application, since $PK_{w0}|PK_{w1}$ is not uniformly distributed when the Lamport scheme is constructed from an arbitrary one-way function. However, the "real" (stronger) definition of UOWHFs in the literature *is* sufficient for the present application. We hope the reader will forgive this simplification.

the *Random Oracle model*, and to prove the security of cryptographic constructions in this model. Although, as we will see, a proof in this model does not guarantee security in the "real world", it still provides a useful check that our construction is not inherently flawed.

In brief (we discuss this in more detail next lecture), the random oracle model assumes the following:

- There is a public oracle that everyone (including all honest parties as well as the adversary) has access to.
- This oracle implements a *truly random function* in the following sense: the first time anyone asks the oracle a query x, it chooses a completely random value y and returns this value. In the future, whenever someone asks query x again, the same answer y is returned. Thus, consistency is maintained. Note that this is equivalent to having the oracle be a "black box" which stores a completely random function (as discussed in an earlier lecture).
- In the real world, of course, random oracles don't exist and even if we wanted to implement a random function we would be unable to for reasonable input/output sizes. Thus, what we do is as follows: design a scheme in the random oracle model and prove its security in that model. Then, to actually use the scheme in the real world, we replace the random oracle by a cryptographic hash function (e.g., SHA-1 or MD5).
- The intuition is that if the hash function is "good" and really "garbles" its input, then it "acts like" a random oracle. Thus, on an intuitive level, a scheme secure in the random oracle model should also be secure in the *standard model* (i.e., the real world, without random oracles) when the random oracle is replaced by a "good" hash function.
- We mention, though, that all the terms of the previous paragraph ("good", "acts like", etc.) are vague; in fact, there is no (known) way to replace a random oracle with *any* hash function so that the resulting scheme is provably secure in the standard model.

References

[1] M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. STOC '89.

[2] J. Rompel. One-Way Functions are Necessary and Sufficient for Secure Signatures. STOC '90.