CAS CS 548. Problem Set 3

Due 5 pm Tuesday, April 18 2006 if your presentation is after 4/24 and 5 pm Friday, April 28, 2006 if your presentation is before 4/21, in the drop box near the CS office.

Problem 1. Consider the Guillou-Quisquater identification scheme (it may help to see Figure 1 in [IR01], where the corresponding signature scheme is described). Add the restriction that $p_1 \equiv p_2 \equiv 3 \pmod 4$. Replace e with 2^m, for some fixed $m \geq l$. Note that it no longer holds that $\gcd(e, \phi(n)) = 1$. Show that the scheme remains a secure identification scheme (note that this means showing some kind of soundness and honest-verifier zero-knowledge properties; soundness will hold under the assumption that taking square roots modulo n is hard, which is equivalent to the assumption that factoring n is hard). Where was the restriction that $p_1 \equiv p_2 \equiv 3 \pmod 4$ used in your argument? Conclude that this scheme's Fiat-Shamir transformation is a secure signature scheme in the random-oracle model (just apply the theorem from [AABN02]).

Problem 2. Consider the following attempt at building a forward-secure signature scheme that uses the ordinary signature scheme from the previous problem as a starting point. Let T be the total number of time periods. Let the public modulus be $n = p_1 p_2$ for $p_1 \equiv p_2 \equiv 3 \pmod 4$, secret key for time period zero be $s \in_R \mathbb{Z}_n^*$, and public key be $v = s^{2^{l+T}}$. To update the key secret key, simply square it modulo n. To sign a message for time period i, use the signature scheme from the previous section with $m = l + T - i$ and make sure to add i to the hash input (note that the verifier knows the correct secret key $s_i = s^{2^i}$ during time period i). There is hope that this is forward-secure: key evolution is clearly one-way (because modular squaring is one-way), and the underlying signature scheme is secure. Show that the scheme is in fact *not* a forward-secure signature scheme.

Problem 3. In class, we showed that pairwise-independent hash functions make good extractors. If we wanted to extract l bits that were ε-close to uniform out of an n-bit string whose minentropy was k, we had to set $l \leq k - 2\log\frac{1}{\varepsilon} + 2$ and we needed a seed of length $n + l$ (for the "chopped-off" $ax + b$ construction: a had to be n bits long, and b had to be l bits long). Note the tradeoff between the quality of bits ε and the number of bits l. Note also that the seed length is linear in n, which means that we need a long seed even when the random input is very poor quality and the minentropy k is much less than n. In this problem you will show that you can have a considerably shorter seed. In fact, the problem of building extractors with short seeds has attracted much attention.

(a) Say that a family of functions $\{H_i\}_{i\in I}$ has collision probability p if for all $x \neq y$, $\Pr_i[H_i(x) = H_i(y)] \leq p$. When $p = 1/|R|$, where $|R|$ is the size of the range, such a family is called *universal*; when $p = (1+\delta)/|R|$, it is called δ-*almost universal*. Note that pairwise-independent functions are, in particular, universal.

Show a universal function family for domain $D = \{0,1\}^n$ and range $\{0,1\}^l$ with seed length n.

(b) Suppose $\{f_i\}_{i\in I}$ is a universal family with domain F and range $\{0,1\}^l$. Suppose $\{g_j\}_{j\in J}$ with domain D and range F has collision probability p. Consider the family $\{f_i \circ g_j\}_{(i,j)\in I\times J}$. How close to uniform are the l bits that this family extracts from any distribution of minentropy k on D?

(c) Let $x = (x_0, \ldots, x_d) \in F^{d+1}$ for some field F. Let $a \in F$. Define $g_a(x) = x_0 + x_1 a + x_2 a^2 + \ldots + x_d a^d$. Show that $\{g_a\}_{a \in F}$ has collision probability $p = d/|F|$.

(d) The trick to getting an extractor with a short seed is to combine the function family from the previous part with a universal function family, as per part (a). The advantage is that the first function family has a short seed, and the second operates only on smaller inputs, and thus can have a shorter seed, as well. There is a tradeoff: we can vary d, thus getting different values for $|F| = |D|^{1/(d+1)}$ and therefore different seed lengths and extractor quality. (Of course, we need F to be a field, and fields exist only for certain sizes, but it won't hurt much to round up $|F|$ to the nearest power of two.)

Show that by setting $d = \frac{n}{k + \log n} - 1$, you will get an extractor whose seed length is at most $2(1 + k + \log n)$ and output length is $l = k - 2 \log \frac{1}{\varepsilon} + 1$. Thus, we extract essentially the same number of bits (just one fewer) and of the same quality, but the seed length depends essentially only on the output length and input entropy.

References

[AABN02] Jee Hea An, Michel Abdalla, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Samir transform: Minimizing assumptions for security and forward-security. In Lars Knudsen, editor, *Advances in Cryptology—EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*. Springer-Verlag, 28 April–2 May 2002.

[IR01] Gene Itkis and Leonid Reyzin. Forward-secure signatures with optimal signing and verifying. In Joe Kilian, editor, *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 332–354. Springer-Verlag, 2001.