6.889: New Developments in Cryptography March 15, 2011

Alternating Extractors and Leakage-Resilient Stream Ciphers

Instructor: Leo Reyzin *Scribe: Drew Wolpert, Sophia Yakoubov*

1 Recap

Definition 1 *If X and Y are two random variables with range U, then the* statistical distance *between X and Y is*
$$\Delta(X,Y) \equiv \frac{1}{2}\sum_{u\in U} |\Pr[X=u]-\Pr[Y=u]|.$$
$X \approx_\varepsilon Y$ is also used to denote $\Delta(X,Y)\le\varepsilon$.

Last time we proved the following lemma:

Lemma 2 *For all randomized functions f, if $x \approx_\varepsilon y$, then $f(x) \approx_\varepsilon f(y)$.*

Definition 3 *A (k,ε)-*extractor *takes a seed U_d that is uniformly distributed on $\{0,1\}^d$, and a random variable X with $H_\infty(X)\ge k$ which is independent of U_d. Then*
$$(Ext(X,U_d),U_d) \approx_\varepsilon (U_l,U_d),$$
where U_l is uniformly distributed on $\{0,1\}^l$, and is independent of X and U_d.

Definition 4 *The* min-entropy *of a discrete random variable X, denoted $H_\infty(X)$, is equal to $-\log(\max(P(X=x)))$.*

Last time we proved the **Leftover Hash Lemma**, which states that if X is a random variable with universe U and $H_\infty(X)\ge k$, $\varepsilon>0$, and $\mathcal{H}$ is a universal hash family of size 2^d with output length $l=k-2\log(1/\varepsilon)$, then
$$\text{Ext}(x,h)=h(x)$$
is a $(k,\varepsilon/2)$ extractor with seed length d and output length m. In other words, $\text{Ext}(x,h)$ extracts l bits from x that are ε-close to uniform, with $\varepsilon=\frac{1}{2}\sqrt{2^{-l}}$. For a fixed ε, the amount of extracted bits l is optimal up to an additive constant, as shown in [5]. (In other words, the loss $2\log(1/\varepsilon)$ bits is necessary.) The only drawback of this extractor is that d (seed length) is high. We need extractors with shorter seeds for the Alternating Extraction game presented in the next section; we will not show how to build them, but they do exist, with seeds as short as $\Theta(\log 1/\varepsilon+\log n)$, where n is the number of bits in elements of X.

2 The Alternating Extraction Theorem

The result in this section is from [2], although the presentation is closer to that of [1]. Let's say we have two players, P_Q and P_W. Player P_Q holds string Q, and player P_W holds string W. Q and W are long, high-entropy, and independent of one-another. P_Q begins the game by sending s_1, which he picks uniformly at random, to P_W. P_W then sends $\text{Ext}(W, s_1)$ back to P_Q. The two players then continue to use extractors on the string they receive, and send the result back, in a game much resembling extractor ping-pong. This looks as follows:

$$\begin{tikzcd}
Q \arrow[r, "s_1"] & W \\
{} & {} \arrow[l, "r_1 = Ext(W; s_1)"'] \\
{} \arrow[r, "s_2 = Ext(Q; r_1)"] & {} \\
{} & {} \arrow[l, "r_2 = Ext(W; s_2)"'] \\
 & \cdots
\end{tikzcd}$$

Note that r_is and s_is can be very short, since they are used only as extractor seeds (and hence can be logarithmic in the length of Q and W). We want to prove that this protocol has a "no peeking ahead feature": that in order to learn s_i and r_i, the players must communicate for i round trips or else essentially send the entire Q or W to the other side. In other words, interaction can logarithmically decrease the amount of communication necessary to learn a certain value; reducing interaction even by a single round will lead to an exponential increase in the number of bits that need to be communicated.

More formally, P_Q and P_W may or may not follow the protocol. Let msg_j^Q be the jth message sent by P_Q. It will contain r_j, as well as any extra information P_Q chose to transmit along with it. msg_j^W is defined symmetrically for P_W. Let V_i^W be the view of P_W up to i; that is, V_i^W is W itself, combined with $(\mathsf{msg}_1^Q, \ldots, \mathsf{msg}_i^Q)$. ($V_i^W$ also contains any random coins P_W tosses internally, if any.) V_i^Q is defined symmetrically for P_Q; however, keep in mind that P_Q is the first player to send a message, so V_i^Q only includes $(\mathsf{msg}_1^W, \ldots, \mathsf{msg}_{i-1}^W)$. The theorem is as follows:

Theorem 5 *Assume the extractor Ext used in the protocol is a (k, ε) average-case extractor (i.e., it produces outputs that are ε-close to uniform as long as the input has average minentropy k). If*

$$\sum_{j=1}^{i} |\mathsf{msg}_j^Q| < H_\infty(Q) - k \quad \text{and} \quad \sum_{j=1}^{i} |\mathsf{msg}_j^W| < H_\infty(W) - k\,,$$

then

$$(V_i^W, s_{i+1}) \approx_{2i\varepsilon} (V_i^W, U) \quad \text{and} \quad (V_i^Q, r_i) \approx_{2i\varepsilon} (V_i^Q, U).$$

In other words, as long as not too much information is sent, s_{i+1} looks uniform to P_W after P_W receives $s_1, \ldots, s_i$ and whatever extra information P_Q chose to send along with it. Symmetrically, r_i looks uniform to P_Q after P_Q receives $r_1, \ldots, r_{i-1}$, and whatever extra information P_W chose to send along with it.

Note that an even stronger statement follows from this theorem; namely, that all future messages r and s values should look uniform to both P_Q and P_W. This corresponds to [1, Theorem 9], whereas what we stated above as Theorem 5, and will proceed to prove, corresponds to [1, Lemma 41]. However, [1, Theorem 9] follows easily from [1, Lemma 41] (our Theorem 5) by the hybrid argument.

Our proof will use the following important fact. Let msgs be the collection of messages sent by both parties from the beginning of the protocol up to some point. We claim that Q and W are independent given msgs; equivalently, that $Q \to \mathsf{msgs} \to W$ form a Markov chain. In other words, if msgs contain information about the joint properties for Q and W (e.g., the exlusive-or of their first bits), the information can be transformed into the piece about Q and a separate piece about W (e.g., the first bit of Q and the first bit of W).

This fact is crucial because the seeds that are used to extract from Q are made out of W, and vice versa, and seeds need to be independent of the source in an extractor.

This fact can be proved by induction. It is easier to think of the following equivalent formulation: it is possible to sample the joint distribution (Q, msgs, W) by sampling $\mathsf{msgs} \to Q$ and $\mathsf{msgs} \to W$. Indeed, the base step is easy: initially Q and W are independent. The inductive step is shown as follows (we'll show it for the message sent by P_W; the other case is symmetric). The message sent by P_W depends only on information available to P_W, i.e., msgs so far and W. Thus, once that message is fixed, the marginal distribution of W changes, but the marginal distribution of Q does not. So sampling $\mathsf{msgs} \to W$ changes in accordance with the newly sent message, but sampling $\mathsf{msgs} \to Q$ remains the same.

Proof: (of Theorem 5) We will prove this by induction; our proof a is summary of the proof in [1], with one correction as pointed out below.

The base case is $i = 0$. s_1 is by definition uniformly random, so it is clear that

$$(V_0^W, s_1) \approx_0 (V_0^W, U).$$

The inductive hypothesis is that

$$(V_{i-1}^W, s_i) \approx_{(2i-2)\varepsilon} (V_{i-1}^W, U).$$

From here we will only do half an iteration - that is, we will show that if the inductive hypothesis holds, then it must be true that

$$(V_i^Q, r_i) \approx_{(2i-1)\varepsilon} (V_i^Q, U).$$

Because of the symmetry of this interaction, half an iteration is sufficient.

The inductive hypothesis gives us that

$$(V_{i-1}^W, s_i) \approx_{(2i-2)\varepsilon} (V_{i-1}^W, U).$$

This implies

$$(V_{i-1}^W, s_i, Ext_W(W, s_i)) \approx_{2(i-1)\varepsilon} (V_{i-1}^W, U, Ext_W(W, U)).$$

This is true by Lemma 2, because the extractor is a function, and we only applied it to things already present on either side (since W is necessarily contained in V_{i-1}^W).

Let $\mathsf{msgs} = ((\mathsf{msg}_1^Q, \ldots, \mathsf{msg}_{i-1}^Q), (\mathsf{msg}_1^W, \ldots, \mathsf{msg}_{i-1}^W))$. Because V_{i-1}^W necessarily contains $(\mathsf{msg}_1^Q, \ldots, \mathsf{msg}_{i-1}^Q)$, and $(\mathsf{msg}_1^W, \ldots, \mathsf{msg}_{i-1}^W)$ can trivially be computed from it, it follows by Lemma 2 that

$$\begin{aligned} & (\mathsf{msgs}, s_i, Ext_W(W, s_i)) \\ \approx_{(2i-2)\varepsilon} \quad & (\mathsf{msgs}, U, Ext_W(W, U)) \,. \end{aligned} \tag{1}$$

Because W has average min-entropy at least k even conditioned on msgs (a rigorous proof of this fact is needed, but intuitively it is guaranteed by the theorem statement's condition on message lengths and by the fact that only messages from P_W can reduce the entropy of W), the extractor should work as expected, and thus by definition of extractor,

$$\begin{aligned} & (\mathsf{msgs}, U, Ext_W(W, U)) \\ \approx_{\varepsilon} \quad & (\mathsf{msgs}, U, U') \,, \end{aligned} \tag{2}$$

where U' is a fresh uniform sample.

Applying the triangle inequality (which holds for statistical distance) to the two statements above, we get

$$\begin{aligned} & (\mathsf{msgs}, s_i, Ext_W(W, s_i)) \\ \approx_{(2i-1)\varepsilon} \quad & (\mathsf{msgs}, U, U') \,. \end{aligned} \tag{3}$$

We now need to replace U with s_i on the right-hand side, because the view V_i^Q contains s_i rather than U. There are two methods to do it. We suggest that a first-time reader simply accept this on faith and skip to the result of this replacement, which is Equation 5.

The first method is relatively simple: we observe that Equation 1 implies (by Lemma 2, where f is the function that simply removes the rightmost term and replaces it with a uniformly random string)

$$\begin{aligned} & (\mathsf{msgs}, U, U') \\ \approx_{2(i-1)\varepsilon} \quad & (\mathsf{msgs}, s_i, U') \,, \end{aligned} \tag{4}$$

and so we can apply the triangle inequality again, this time to equations 3 and 4. Unfortunately, the two applications of the triangle inequality will double the statistical distance, and the end result will be that statistical distance will grow exponentially, rather than linearly, with i. That's not going to change the result dramatically (we can simply set k high enough and use good enough Ext that ε is small to begin with), but it's better not to have to do this. So we will have to use different, more complicated method.

The more complicated method was described (with mistakes) in [2, Lemma 1] and [1, Lemma 31]; this description fixes those mistakes, and is based on private communication with Stefan Dziembowski.

Lemma 6 *Suppose A, B, C, D, and F are random variables such that C and D vary over the same domain, and*

- *C is independent of F given A*
- *C is independent of B given A*
- *D is independend of F given A*

- *D is independend of B given A*

(in other words, $C \to A \to F$, $C \to A \to B$, $D \to A \to F$, and $D \to A \to B$ form Markov chains). Suppose f is a function such that $(A, C, f(B, C)) \approx_{\delta_1} (A, C, F)$ and $(A, C) \approx_{\delta_2} (A, D)$. Then

$$(A, D, f(B, D)) \approx_{\delta_1+\delta_2} (A, D, F)\,.$$

Proof: First, to simplify notation, fix a paricular value of A and drop A from all formulas; at the end, simply average over all values A. By independence of D from B and F,

$$\Delta((D, f(B, D)), (D, F)) = \sum_x \Delta(f(B, x), F) \Pr[D = x]\,.$$

This, in turn, is equal to

$$\sum_x \Delta(f(B, x), F)(\Pr[D = x] - \Pr[C = x]) + \sum_x \Delta(f(B, x), F) \Pr[C = x])\,.$$

The second term is simply δ_1 (by independence of C from B and F). The first term is at most δ_2, by definition of statistical distance. □

To apply the above lemma, let

- f be the extractor function,
- $A = \mathsf{msgs}$,
- $B = W$,
- $C = U$,
- $D = s_i$,
- $F = U'$.

Since F and C are uniform and independent, the first three Markov chain conditions are trivially satisfied. The fourth Markov chain condition follows from the fact that Q, msgs, and W form a Markov chain and $D = s_i$ is a just a function of Q and msgs. The remaining conditions of the lemma are satisfied by equations 2 and 1. It follows that

$$\begin{aligned} & (\mathsf{msgs}, s_i, Ext_W(W, s_i)) \\ \approx_{(2i-1)\varepsilon}\ & (\mathsf{msgs}, s_i, U')\,. \end{aligned} \tag{5}$$

There is one step left to get to our desired result.

As a consequence of $Q \to \mathsf{msgs} \to Q$ being a Markov chain, we claim that there exists a randomized function f such that the joint distribution $(Q, \mathsf{msgs}, s_i, W)$ is identical to the joint distribution of $(f(\mathsf{msgs}, s_i), \mathsf{msgs}, s_i, W)$. (In other words, Q can be obtained by a random function of (msgs, s_i) and will be distributed correctly even in the presence of W.) By Lemma 2, we get

$$\begin{aligned} & (Q, \mathsf{msgs}, s_i, Ext_W(W, s_i)) \\ \approx_{(2i-1)\varepsilon}\ & (Q, \mathsf{msgs}, s_i, U')\,. \end{aligned}$$

Note that it is crucial here that W is not needed to sample Q, since W is present in the left-hand side of the equation, but not in the right-hand side. Note also that it is crucial that we managed to get s_i into both sides of the equations (in other words, Equation 3 is insufficient), since otherwise on the left-hand side we would have to sample to Q in the presence of s_i and on the right-hand side we would have to sample Q in the absence of s_i, which would mean that we are applying different functions f to the two sides.

Since V_i^Q is just Q and $(\mathsf{msg}_1^W, \ldots, \mathsf{msg}_{i-1}^W)$, and $r_i = Ext_W(W, s_i)$, it follows that

$$\Rightarrow (V_i^Q, r_i) \approx_{(2i-1)\varepsilon} (V_i^Q, U')$$

Which is what we wanted to show. □

3 Applications

In this section we'll discuss one application of what we just learned, to leakage-resilient cryptography.

3.1 Constructing a Leakage-Resilient Pseudorandom Generator

First, a note about leakage! It's all well and good to prove theorems about cryptographic schemes, but the traditional ones don't consider the innards of the device that uses the secret key. It's flashing and blinking and whirring and sending all kinds of signals – can *these* tell the adversary something we don't want him to know? Yes! They convey all the information an expert listener needs. If you listen closely to, for example, an RSA private key operation, you can hear squaring and multiplication as they happen (and they sound different), so you can actually read off the secret key just from the noises the decryption algorithm makes [6]. In fact, power consumption,timing, EM radiation, etc., can all leak information about the secret key.

So we need some assumptions: first, let us assume that there is in fact some secure memory (unfortunate adage: "only computation leaks information"). So you can *move* and *erase* things, and you can leave them places (pouring the sand into the bottle is leaky, but once it's in and corked, it is secure). In our model, then, the adversary can get information out of whatever you are working on right now.

We will now present a stream cipher (psuedorandom generator) that is leakage resilient. The result is from [3] These are our assumptions:

1. There is a way store (put away) bits so that they don't leak. However, they cannot be operated upon when so stored.
2. Leakage per unit time is bounded.
3. We have secure erasure: working memory forgets its previous states.

Our goal is to produce a pseudorandom generator (PRG) that outputs blocks such that $block_i \approx U$ even to an adversary who knows all of $block_1 \ldots block_{i-1}$ and was allowed to observe their computation.

We have already almost done this. Recall the extractor ping-pong game we defined in the first section, and this picture:

$$
\begin{array}{ccc}
Q & \xrightarrow{\quad s_1 \quad} & W \\
 & \xleftarrow{\quad r_1 = Ext(W; s_1) \quad} & \\
 & \xrightarrow{\quad s_2 = Ext(Q; r_1) \quad} & \\
 & \xleftarrow{\quad r_2 = Ext(W; s_2) \quad} & \\
 & \ldots &
\end{array}
$$

During each computation, we engage the middle (i.e., the r_i or s_i) and *one of* the left or the right side (i.e., Q or W), but never both sides at once. So, we can put the part we are not working on into nonleaky memory, and, under our assumption, the leakage will depend on (Q, r_i) or on (W, s_i). In this case, to say that the adversary gets a bounded amount of leakage is the same as to saying that P_W or P_Q sends a bounded amount of extra information; and we know by Theorem 5 that the next message looks uniformly random even with that extra information. So this construction, now computed by a single computer instead of two players (but with memory separated into components that never leak together), works as a leakage-resilient PRG: the messages exchanged—the r_is and the s_is—will simply be the outputs.

This works until we leak/exchange enough information to run out of entropy in Q and W. But that's, of course, not very interesting, because the whole point of a PRG is to output more randomness that its input. So we need to apply a psuedorandom generator to Q or W at each round, in order to add to computational entropy. Note that pseudorandom generators should be applied to uniform seeds, so we will first apply an extractor to condense the entropy of what's remaining.

$$
\begin{array}{lcl}
Q & \xrightarrow{\qquad\quad s_1 \qquad\quad} & W \\
 & & (r_1, X_1) = Ext(W; s_1) \\
 & & W_2 = PRG(X_1) \\
 & \xleftarrow{\qquad\quad r_1 \qquad\quad} & \\
(s_2, Y_2) = Ext(Q; r_1) & & \\
Q_2 = PRG(Y_2) & & \\
 & \xrightarrow{\qquad\quad s_2 \qquad\quad} & \\
 & & (r_2, X_2) = Ext(W_2; s_2) \\
 & & W_3 = PRG(X_2) \\
 & \xleftarrow{\qquad\quad r_2 \qquad\quad} & \\
(s_3, Y_3) = Ext(Q_2; r_2) & & \\
Q_3 = PRG(Y_3) & & \\
 & \ldots &
\end{array}
$$

The result of the previous section worked (i.e., produced uniform outputs) as long as Q and W had enough min-entropy: the Q_i and W_i eventually won't (because we will be running the PRG for longer than the amount of entropy we have). We need to show that they are indistinguishable from something that does, so nobody can tell the difference (indistinguishable!), and thus the result from the previous section can still be used to get pseudorandom, rather than uniform, outputs.

Note that Q_is andW_is are the result of applying a PRG to a uniform string, so it should all work. The only problem is that the PRG and the extractor used to get Q_i and W_i leak during their operation, so Q_i and W_i are not pseudorandom any more. Our only hope is that they have something analogous to min-entropy—i.e., are indistinguishable from something that has min-entropy from the point of view of a computationally bounded adversary.

3.2 `HILL` entropy

Definition 7 *A distribution X has* `HILL` *entropy k if there exists a distribution Z such that:*

1. *$H_\infty(Z) \geq k$ and*
2. *For every distinguisher D, circuit of size $\leq S$, $E(D(X)) - E(D(Z)) \leq \varepsilon$*

Then we say that $H^{\texttt{HILL}}_{\varepsilon,S}(X) \leq k$.

Why do we need this? Because our hope is Q_i and W_i have HILL entropy. And, from the point of view of a computationally bounded adversary, HILL entropy is as good as real min-entropy, so if our hope is right, we are done.

We know outputs of a PRG are indistinguishable from random, and thus have as much HILL entropy as their length. But what about outputs of PRG that was leaky?

More generally, suppose (X, Y) is a correlated pair and $H^{HILL}(X) \geq k$. What about $H^{\texttt{HILL}}(X|Y)$? (In our application, X is the PRG output and Y is the leakage.) Once we figure that out, we'll be done.

First let's make a pretty statement:

$$H^{\texttt{Metric}^*}_{\varepsilon|Y|,S}(X|Y) \geq H^{\texttt{Metric}^*}_{\varepsilon,S}(X) - \log|\operatorname{supp}(Y)| \tag{6}$$

.

Unfortunately, this is known for `Metric`* but *not* for `HILL` entropy, and `Metric`* is ugly. However:

$$H^{\texttt{Metric}^*}_{\delta,S}(X|Y) \geq k \implies H^{HILL}_{2\delta,\Omega(\delta^2 s/|X|)}(X|Y) \geq k \tag{7}$$

and conversely:

$$H^{\texttt{HILL}}_{\delta,S}(X|Y) \geq k \implies H^{\texttt{Metric}^*}_{\delta,S}(X|Y) \geq k \tag{8}$$

So what is this `Metric`* entropy?

Definition 8 *X has* `Metric`* *entropy k if for every D there exists a Z such that:*

1. *$H_\infty(Z) \geq k$ and*
2. *For every distinguisher D, circuit of size $\leq S$, $E(D(X)) - E(D(Z)) \leq \varepsilon$*

as in the definition of HILL entropy, but with the further restriction that we consider only deterministic D outputting values in the range $[0, 1]$. Then we say that $H^{metric*}_{\varepsilon,S}(X) \leq k$.

Now let's prove the pretty statement (6). Actually, we will prove:

Theorem 9
$$H^{metric*}(X|Y=y) \geq H^{metric*}_{\varepsilon,S}(X) - \log\frac{1}{Pr[Y=y]} \tag{9}$$

which implies (6).

Proof sketch (for full proof see [4]):

1. Suppose D distinguishes $X|Y=y$ from any distribution Z of min-entropy $\nu-\Delta$ with advantage ε'. Show that either for all such Z, $\mathbb{E}[D(Z)]$ is lower than $\mathbb{E}[D(X|Y=y)]$ by at least ε', or for all such Z, $\mathbb{E}[D(Z)]-\varepsilon'$ is higher than $\mathbb{E}[D(X|Y=y)]$ by at least ε'. Without loss of generality, let us assume the former, which allows us to remove absolute values and to find a high-entropy distribution Z^+ on which $\mathbb{E}[D(Z)]$ is the highest.
2. Show that there exists a distinguisher D' that also has advantage ε' but outputs only 0 or 1. This can be done by setting a cutoff: if D's output is above the cutoff, it will output 1, otherwise it will output 0.
3. Show that for every every z outside of Z^+, D' outputs 0, and that Z^+ is essentially flat. Use these two facts to show an upper bound on $\mathbb{E}[D'(W)]$ for any W of min-entropy ν.
4. Show a lower bound on $\mathbb{E}[D'(X)]$.

Suppose not, and $H^{Metric^*}_{\varepsilon|Y|,S}(X|Y) < H^{metric*}_{\varepsilon,S}(X) - \log|\operatorname{supp}(Y)|$. This means there is a single distinguisher D that works for all distributions. Let $\beta = E[D(X|Y=y)]$, $\varepsilon' = \frac{\varepsilon}{Pr[Y=y]}$.

Suppose $H^{Metric^*}_{\varepsilon|Y|,S}(X) \geq v$ for some v. Let $\varepsilon' := \varepsilon/Pr[Y=y]$ and χ be the outcome space of X. Assume for contradiction that
$$H^{Metric^*}_{\varepsilon',S}(X|Y) \geq v - \log\frac{1}{Pr[Y=y]} \tag{10}$$

does not hold. Then by the definition of metric entropy, there exists a distinguisher D that works for all distributions. So for all Z with $H_\infty(Z) \geq v - \log\frac{1}{Pr[Y=y]}$,
$$|\mathbb{E}[D_y(X)|Y=y]| - \mathbb{E}[D_y(Z)] > \varepsilon' \tag{11}$$

Let Z^+ and Z^- be distributions of min-entropy $v - \log(1/Pr[Y=y])$ that maximize and minimize the expectations $\beta^+ = \mathbb{E}[D_y(Z^+)]$ and $\beta^- = \mathbb{E}[D_y(Z^-)]$ respectively.

References

[1] Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *STOC, Symposium on Theory of Computing*, 2009.

[2] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret-sharing. In *FOCS, 48th Annual IEEE Symposium on Foundations of Computer Science*, 2007.

[3] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS, 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008.

[4] Benjamin Fuller and Leonid Reyzin. Computational entropy and information leakage. `http://cs-people.bu.edu/bfuller/metricEntropy.pdf`, 2011.

[5] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[6] Eran Tromer and Adi Shamir. Acoustic cryptanalysis: on nosy people and noisy machines. Presented at Eurocrypt 2004 Rump Session. See `http://people.csail.mit.edu/tromer/acoustic/`, 2004.