Topics in Algebra: Cryptography

Univ.-Prof. Dr. Goulnara ARZHANTSEVA

WS 2018

Pseudorandomness

Cryptography:

- Symmetric and asymmetric cryptosystems;
- One-way functions, Hash functions;
- Key management, Digital Signatures, Applications;
- Pseudorandom generators.

Encoding, Error-correction.

Randomness vs Pseudorandomness

Random numbers	Pseudorandom number
Nondeterministic	Deterministic
Physical processes, hardware	Computer algorithm, software
No pattern	Periodic
Unpredictable	Predictable, depending on observers

Two of the most celebrated open problems in mathematics and computer science, the Riemann Hypothesis and the P vs. NP question, can be stated as problems about pseudorandomness.

Bit generator

A seed is a number (or a vector) used to initialize a pseudorandom number generator.

Definition: (k, l)-bit generator

$k, l \in \mathbb{N}, l \geqslant k + 1$. A (k, l)-bit generator is

$$f : (\mathbb{Z}_2)^k \to (\mathbb{Z}_2)^l$$

that is in P (as a function of k).

Bit generator

A seed is a number (or a vector) used to initialize a pseudorandom number generator.

Definition: (k, l)-bit generator

$k, l \in \mathbb{N}, l \geqslant k + 1$. A (k, l)-bit generator is

$$f : (\mathbb{Z}_2)^k \to (\mathbb{Z}_2)^l$$

that is in P (as a function of k).

The input $s_0 \in (\mathbb{Z}_2)^k$ is the seed, and the output $f(s_0) \in (\mathbb{Z}_2)^l$ is the generated bitstring.

We assume that l is a polynomial function of k, called the stretch function of f.

Bit generator

A bit generator is deterministic.

We aim to construct bit generators so that $f(s_0)$ looks like random bits. Such a bit generator is called a pseudo-random bit generator (PRBG).

Example of use: A seed is a secrete key, and a bit-generator generates a key of the same length as the plaintext for the one-time pad.

Linear Feedback Shift Register: Definition

Definition: LFSR for $c = (c_0, \ldots, c_{l-1})^T \in (\mathbb{Z}_2)^l$ of degree $l > 0$, $c_0 \neq 0$

It is given by the linear recurrence:

$$s_{n+l} = (s_n, \cdots, s_{n+l-1}) \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_{l-1} \end{pmatrix} \quad n \geqslant 0,$$

such that
$t^{(0)} := t = (t_0, \cdots, t_{l-1}) \in (\mathbb{Z}_2)^l$ is the initial value,
$s_i = t_i$ for $0 \leqslant i \leqslant l-1$,
$t^{(n)} := (s_n, \cdots, s_{n+l-1})$ is the n-th state vector.
We write $s := \langle c, t \rangle$.

It is of degree l as each term depends on the previous l terms.

Question 23: Why $c_0 \neq 0$?

Linear Feedback Shift Register: Example

LFSR for $c = (1, 1, 0, 0)^T \in (\mathbb{Z}_2)^4$ of degree $l = 4$ with $t = (1\ 0\ 1\ 0)$

$$s = \underline{1, 0, 1, 0}, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, \mid \underline{1, 0, 1, 0}, \ldots$$

Linear Feedback Shift Register: Example

LFSR for $c = (1, 1, 0, 0)^T \in (\mathbb{Z}_2)^4$ of degree $l = 4$ with $t = (1\ 0\ 1\ 0)$

$$s = \underline{1, 0, 1, 0}, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, \mid \underline{1, 0, 1, 0}, \ldots$$

Definition: periods of LFSR

s is k-periodic if $s_{i+k} = s_i \ \forall i \geqslant 0$, or equivalently, $t^{(i+k)} = t^{(i)} \ \forall i \geqslant 0$.
c is k-periodic if $s = \langle c, t \rangle$ is k-periodic for all $t \in (\mathbb{Z}_2)^l$.
The period is the smallest such number k.

Take $k = 2^l - 1$. So, a short initial 'key' (seed) generates a keystream with a long period:
given an l-bit seed, an LFSR of degree l produces $2^l - l - 1$ further bits before repeating.

Question 24: Is this k the period?

Linear Feedback Shift Register: Security

The LFSR is insecure! The knowledge of any $2l$ consecutive bits allows to determine the seed, and hence the entire sequence.

For each $n \geqslant 0$, the linear recurrence expressing s_{n+l} is a linear equation in the l unknowns $(c_0, \ldots, c_{l-1})$. For $n \in \{0, 1, \ldots, l-1\}$, we get l linear equations in l unknowns:

$$(s_l, s_{l+1}, \ldots, s_{2l-1}) = (c_0, c_1, \cdots, c_{l-1}) \cdot \begin{pmatrix} s_0 & s_1 & \ldots & s_{l-1} \\ s_1 & s_2 & \ldots & s_l \\ \vdots & \vdots & & \vdots \\ s_{l-1} & s_l & \ldots & s_{2l-2} \end{pmatrix}$$

If the matrix has the inverse $\mod 2$, then we find $(c_0, \ldots, c_{l-1})$ and determine the entire sequence.

Test question

Question 25

Show that the $l \times l$ coefficient matrix from the previous slide is indeed invertible mod 2.

Hint: let $v_i = (s_i, \ldots, s_{i+l-1})$ for $i \geqslant 0$. The coefficient matrix has $v_0, \ldots v_{l-1}$ as rows. The goal is to prove that these l vectors are linearly independent.

Remark: the coefficient matrix is an example of a Hankel matrix.

A bit generator: Example

An LFSR of degree l is an example of a bit-generator.

Question 26

Consider an LFSR as a bit generator, what are, in this case, parameters k and l from the definition of bit-generator?

An RSA bit generator

Definition: RSA generator

Let p, q be $k/2$-bit primes, $n = pq$. Let e be such that $\gcd(e, \phi(n)) = 1$. A seed s_0 is any element of $(\mathbb{Z}_n)^\times$, so it has k bits. For $i \geqslant 1$, we define

$$s_{i+1} = s_i^e \bmod n,$$

and then we define

$$f(s_0) = (z_1, z_2, \ldots, z_l),$$

where $z_i = s_i \bmod 2, 1 \leqslant i \leqslant l$. Then f is a (k, l)-RSA generator.

Public-key is (n, e) and private-key is (p, q).

Assumption: the Factoring is not in BPP.

Towards a pseudo-random number generator

A pseudo-random number generator should be fast (i.e. computable in polynomial time) and secure.

Our examples are fast. How secure they are?

Intuitively: it should be impossible in an amount of time that is polynomial in k (equivalently, polynomial in l) to distinguish a string of l bits produced by a PRBG from a string of l truly random bits.

Towards a pseudo-random number generator

Example: if a bit generator produces 1 with probability 2/3, then on average a generated bitstring of length l will contain $2l/3$ bits 1.

In contrast, a truely random bitstring of length l will contain $l/2$ 1's on average.

Given a bitstring with l_1 1's, if $l_1 > \frac{l/2+2l/3}{2} = \frac{7l}{12}$, then we conclude that it is a generated bitstring (not a truely random).

Deterministic distinguisher

Notation: $z^i = (z_1, \ldots, z_i)$

Definition: Distinguisher

Let p_0 and p_1 be two probability distributions on $(\mathbb{Z}_2)^l$. For $j = 0, 1$ and $z^l \in (\mathbb{Z}_2)^l$ we denote by $p_j(z^l)$ the probability that the string z^l occurs in the distribution p_j. Let $\mathrm{dst}\colon (\mathbb{Z}_2)^l \to \{0, 1\}$ be a function and $\epsilon > 0$. We define

$$\mathbb{E}_{\mathrm{dst}}(p_j) = \sum_{\{z^l \in (\mathbb{Z}_2)^l \,:\, \mathrm{dst}(z^l)=1\}} p_j(z^l).$$

We say that dst is an ϵ-distinguisher of p_0 and p_1 provided that

$$|\mathbb{E}_{\mathrm{dst}}(p_0) - \mathbb{E}_{\mathrm{dst}}(p_1)| \geqslant \epsilon,$$

p_0 and p_1 are ϵ-distinguishable if there exists an ϵ-distinguisher of p_0 and p_1.

If $\mathrm{dst}(z^l)$ can be computed in polynomial time, it is a polynomial-time distinguisher.

Randomized distinguisher

As above but with

$$\mathbb{E}_{\mathrm{dst}}(p_j) = \sum_{z^l \in (\mathbb{Z}_2)^l} p_j(z^l) \cdot \Pr[\,\mathrm{dst}(z^l) = 1].$$

Towards a pseudorandom generator

A truly random sequence corresponds to the uniform distribution p_{u_l} on the set of all bitstrings of length l:
each string among all 2^l strings can occur with probability $1/2^l$.
If f is a bit generator with a k-bit seed chosen uniformly at random, then we obtain a probability distribution $p_f = f(p_{u_k})$ on the same set.

Towards a pseudorandom generator

A truly random sequence corresponds to the uniform distribution p_{u_l} on the set of all bitstrings of length l:
each string among all 2^l strings can occur with probability $1/2^l$.
If f is a bit generator with a k-bit seed chosen uniformly at random, then we obtain a probability distribution $p_f = f(p_{u_k})$ on the same set.

p_f is very non-uniform

If we assume that no two seeds give same sequence of bits. Then, of the 2^l possible sequences, 2^k sequences each occur with probability $1/2^k$, and the remaining $2^l - 2^k$ sequences never occur.

We would like to have f such that p_{u_l} and p_f are ϵ-distinguishable in polynomial time only for small values of ϵ.

Towards a pseudorandom generator

A truly random sequence corresponds to the uniform distribution p_{u_l} on the set of all bitstrings of length l:
each string among all 2^l strings can occur with probability $1/2^l$.
If f is a bit generator with a k-bit seed chosen uniformly at random, then we obtain a probability distribution $p_f = f(p_{u_k})$ on the same set.

p_f is very non-uniform

If we assume that no two seeds give same sequence of bits. Then, of the 2^l possible sequences, 2^k sequences each occur with probability $1/2^k$, and the remaining $2^l - 2^k$ sequences never occur.

We would like to have f such that p_{u_l} and p_f are ϵ-distinguishable in polynomial time only for small values of ϵ.

Exercise: producing 0's and 1's with equal probability is not sufficient to ensure indistinguishability.

Next bit predictor

Let f be a (k, l)-bit generator.

Definition: Next bit predictor

Let $1 \leqslant i \leqslant l - 1$. A next bit predictor for f is a function

$$\mathrm{nbp} \colon (\mathbb{Z}_2)^{i-1} \to \mathbb{Z}_2,$$

which takes as input an $(i-1)$-tuple $z^{i-1} = (z_1, \ldots, z_{i-1})$, the first $i-1$ bits produced by f (given, an unknown, truly random, k-bit seed), and produces by a polynomial time probabilistic algorithm, the ith bit of the bitstring generated by f (given the first $i-1$ bits) with probability at least $1/2 + \epsilon$, where $\epsilon > 0$.

Next bit predictor: Theorem

p_f induces the probability distribution on any of the l generated bits (or on any subsequence of these l generated bits).

For $1 \leqslant i \leqslant l$, we think of the *i*th generated bit as a random variable $\mathbf{z}_i$.

Theorem: Next bit predictor

Let f be a (k, l)-bit generator. Then the nbp is an ϵ-ith bit predictor for f if and only if

$$\sum_{z^{i-1} \in (\mathbb{Z}_2)^{i-1}} p_f(z^{i-1}) \cdot \Pr\left[\mathbf{z}_i = \mathrm{nbp}(z^{i-1}) \mid z^{i-1}\right] \geqslant \frac{1}{2} + \epsilon.$$

Next bit predictor: a straightforward proof

Proof:

The probability of correctly predicting the ith generated bit, $\Pr[\mathbf{z}_i = \mathrm{nbp}(z^{i-1})]$, is computed by summing over all possible $(i-1)$-tuples $z^{i-1} = (z_1, \ldots z_{i-1})$ the product of the probability that the $(i-1)$-tuple z^{i-1} is produced by the bit generator f and the probability that the ith bit is predicted correctly, given the $(i-1)$-tuple z^{i-1}. ■

Main result

Main result: A next bit predictor is a universal test

A bit generator is secure if and only if there does not exist any polynomial-time ϵ-ith bit predictor for the generator, except for very small values of ϵ.

One direction of the implication is given by the next result.
Here, **Dist** has z^i as an input, and 1 as output if the value predicted by $\mathrm{nbp}(z^{i-1})$ is the same as the actual value of z_i. Otherwise, it outputs 0.

Theorem: from nbp to distinguisher

Let nbp be a polynomial time ϵ-ith bit predictor for the (k, l)-bit generator f, and p_f, p_{u_l} be as above, on $(\mathbb{Z}_2)^i$. Then the distinguisher algorithm **Dist** is a polynomial-time ϵ-distinguisher of p_f and p_{u_l}.

Theorem: from nbp to distinguisher

Proof: By definition, $\mathbf{Dist}(z^i) = 1 \iff \mathrm{nbp}(z^{i-1}) = z_i$. Then,

$$\mathbb{E}_{\mathbf{Dist}}(p_f) = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathbf{Dist}(z^i) = 1] = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = z_i]$$

Theorem: from nbp to distinguisher

Proof: By definition, $\mathbf{Dist}(z^i) = 1 \iff \mathrm{nbp}(z^{i-1}) = z_i$. Then,

$$\mathbb{E}_{\mathbf{Dist}}(p_f) = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathbf{Dist}(z^i) = 1] = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = z_i]$$

Define $z = (z_1, \ldots, z_{i-1}, 0)$ and $z' = (z_1, \ldots, z_{i-1}, 1)$. Then,

$$p_f(z) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 0] + p_f(z') \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 1] =$$

Theorem: from nbp to distinguisher

Proof: By definition, $\mathbf{Dist}(z^i) = 1 \iff \mathrm{nbp}(z^{i-1}) = z_i$. Then,

$$\mathbb{E}_{\mathbf{Dist}}(p_f) = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathbf{Dist}(z^i) = 1] = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = z_i]$$

Define $z = (z_1, \ldots, z_{i-1}, 0)$ and $z' = (z_1, \ldots, z_{i-1}, 1)$. Then,

$$p_f(z) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 0] + p_f(z') \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 1] =$$

$$p_f(z^{i-1}) \cdot \sum_{j \in \{0,1\}} \Pr[\mathbf{z}_i = j \mid z^{i-1}] \cdot \Pr[\mathrm{nbp}(z^{i-1}) = j] =$$

Theorem: from nbp to distinguisher

Proof: By definition, $\mathbf{Dist}(z^i) = 1 \iff \mathrm{nbp}(z^{i-1}) = z_i$. Then,

$$\mathbb{E}_{\mathbf{Dist}}(p_f) = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathbf{Dist}(z^i) = 1] = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = z_i]$$

Define $z = (z_1, \ldots, z_{i-1}, 0)$ and $z' = (z_1, \ldots, z_{i-1}, 1)$. Then,

$$p_f(z) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 0] + p_f(z') \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 1] =$$

$$p_f(z^{i-1}) \cdot \sum_{j \in \{0,1\}} \Pr[\mathbf{z}_i = j \mid z^{i-1}] \cdot \Pr[\mathrm{nbp}(z^{i-1}) = j] =$$

$$p_f(z^{i-1}) \cdot \Pr[\mathbf{z}_i = \mathrm{nbp}(z^{i-1}) \mid z^{i-1}].$$

Theorem: from nbp to distinguisher

Proof: By definition, $\mathbf{Dist}(z^i) = 1 \iff \mathrm{nbp}(z^{i-1}) = z_i$. Then,

$$\mathbb{E}_{\mathbf{Dist}}(p_f) = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathbf{Dist}(z^i) = 1] = \sum_{z^i \in (\mathbb{Z}_2)^i} p_f(z^i) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = z_i]$$

Define $z = (z_1, \ldots, z_{i-1}, 0)$ and $z' = (z_1, \ldots, z_{i-1}, 1)$. Then,

$$p_f(z) \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 0] + p_f(z') \cdot \Pr[\mathrm{nbp}(z^{i-1}) = 1] =$$

$$p_f(z^{i-1}) \cdot \sum_{j \in \{0,1\}} \Pr[\mathbf{z}_i = j \mid z^{i-1}] \cdot \Pr[\mathrm{nbp}(z^{i-1}) = j] =$$

$$p_f(z^{i-1}) \cdot \Pr[\mathbf{z}_i = \mathrm{nbp}(z^{i-1}) \mid z^{i-1}].$$

It follows that

$$\mathbb{E}_{\mathbf{Dist}}(p_f) = \sum_{z^{i-1} \in (\mathbb{Z}_2)^{i-1}} p_f(z^{i-1}) \cdot \Pr[\mathbf{z}_i = \mathrm{nbp}(z^{i-1}) \mid z^{i-1}] \geqslant \frac{1}{2} + \epsilon,$$

as nbp is an ϵ-ith bit predictor (use the previous Theorem).

Theorem: from nbp to distinguisher (suite)

On the other hand, any predictor will predict the ith bit of a truly random sequence with probability $1/2$. Therefore, $\mathbb{E}_{\mathbf{Dist}}(p_{u_l}) = 1/2$. Hence,

$$|\mathbb{E}_{\mathbf{Dist}}(p_{u_l}) - \mathbb{E}_{\mathbf{Dist}}(p_f)| \geqslant \epsilon.$$

as required. ■

Main theorem

Theorem: from distinguisher to nbp Yao'1982

Suppose dst is a (polynomial-time) ϵ-distinguisher of p_f and p_{u_l}, where p_f is the probability distribution induced on $(\mathbb{Z}_2)^l$ by the (k, l)-bit PRBG f, and p_{u_l} is the uniform probability distribution on $(\mathbb{Z}_2)^l$. Then for some $i, 1 \leqslant i \leqslant l-1$, there exists a polynomial-time ϵ/l-ith bit predictor for f.

That is, a pseudo-random bit generator is secure if there does not exist an ϵ-next bit predictor except for very small values of ϵ.

Main theorem: proof

Proof: (Hybrid argument) For $0 \leqslant i \leqslant l$, let q_i be a probability distribution on $(\mathbb{Z}_2)^l$ with first i bits generated by f, and the other $l - i$ bits are generated truly randomly. Thus, $q_0 = p_{u_l}$ and $q_l = p_f$.

By hypothesis, $|\mathbb{E}_{\text{dst}}(q_0) - \mathbb{E}_{\text{dst}}(q_l)| \geqslant \epsilon$. By the triangle inequality,

$$|\mathbb{E}_{\text{dst}}(q_0) - \mathbb{E}_{\text{dst}}(q_l)| \leqslant \sum_{i=1}^{l} |\mathbb{E}_{\text{dst}}(q_{i-1}) - \mathbb{E}_{\text{dst}}(q_i)|.$$

Then there is i, $1 \leqslant i \leqslant l$, such that $|\mathbb{E}_{\text{dst}}(q_{i-1}) - \mathbb{E}_{\text{dst}}(q_i)| \geqslant \frac{\epsilon}{l}$. WLOG, we assume

$$\mathbb{E}_{\text{dst}}(q_{i-1}) - \mathbb{E}_{\text{dst}}(q_i) \geqslant \frac{\epsilon}{l}.$$

We will construct an ϵ-ith bit predictor for this value of i.

Main theorem: proof (continued)

Intuitively: The predicting algorithm produces an l-tuple according to q_{i-1}, given that z^{i-1} is generated by the PRBG. If dst answers 0, then it thinks that the l-tuple was generated according to q_i.

The ith bit is truly random in q_{i-1}, it is given by the PRBG in q_i.

Hence, if dst answers 0, it thinks that the ith bit, z_i is what would be produced by the PRBG. Then z_i is our prediction for the ith bit.

If dst answers 1, it thinks that z_i is truly random, so we take $1 - z_i$ as our prediction for the ith bit.

Main theorem: proof (continued)

Intuitively: The predicting algorithm produces an l-tuple according to q_{i-1}, given that z^{i-1} is generated by the PRBG. If dst answers 0, then it thinks that the l-tuple was generated according to q_i.

The ith bit is truly random in q_{i-1}, it is given by the PRBG in q_i.

Hence, if dst answers 0, it thinks that the ith bit, z_i is what would be produced by the PRBG. Then z_i is our prediction for the ith bit.

If dst answers 1, it thinks that z_i is truly random, so we take $1 - z_i$ as our prediction for the ith bit.

`Input:` $z^{i-1} = (z_1, \ldots, z_{i-1})$
`Choose` $(z_i, \ldots, z_l) \in (\mathbb{Z}_2)^{l-i+1}$ `truly randomly`
`Compute` $z = \mathrm{dst}(z_1, \ldots, z_l)$
`Define` $\mathrm{nbp}(z_1, \ldots, z_{i-1}) = (z + z_i) \bmod 2$

Main theorem: proof (continued)

If dst gives 0, then the prediction is correct with probability $p_f(z_i \mid z^{i-1})$

If dst gives 1, then it is correct with probability $1 - p_f(z_i \mid z^{i-1})$.

Let $\mathbf{z} = z^l$. We have

$$q_{i-1}(\mathbf{z}) \cdot p_f(z_i \mid z^{i-1}) = q_i(\mathbf{z})/2.$$

Main theorem: proof (continued)

$$\Pr[\mathbf{z}_i = \mathrm{nbp}(z^{i-1})] =$$

$$\sum_{\mathbf{z}\in(\mathbb{Z}_2)^l} q_{i-1}(\mathbf{z}) \left(\Pr[\mathrm{dst}(\mathbf{z}) = 0] \cdot p_f(z_i \mid z^{i-1}) + \Pr[\mathrm{dst}(\mathbf{z}) = 1] \cdot (1 - p_f(z_i \mid z^{i-1}))\right) =$$

$$\sum_{\mathbf{z}\in(\mathbb{Z}_2)^l} \frac{q_i(\mathbf{z})}{2} \cdot \Pr[\mathrm{dst}(\mathbf{z}) = 0] + \sum_{\mathbf{z}\in(\mathbb{Z}_2)^l} q_{i-1}(\mathbf{z}) \cdot \Pr[\mathrm{dst}(\mathbf{z}) = 1] - \sum_{\mathbf{z}\in(\mathbb{Z}_2)^l} \frac{q_i(\mathbf{z})}{2} \cdot \Pr[\mathrm{dst}(\mathbf{z}) = 1]$$

$$= \frac{1 - \mathbb{E}_{\mathrm{dst}}(q_i)}{2} + \mathbb{E}_{\mathrm{dst}}(q_{i-1}) - \frac{\mathbb{E}_{\mathrm{dst}}(q_i)}{2} = \frac{1}{2} + \mathbb{E}_{\mathrm{dst}}(q_{i-1}) - \mathbb{E}_{\mathrm{dst}}(q_i) \geqslant \frac{1}{2} + \frac{\epsilon}{l}.$$

Main theorem: Summary

The ϵ-distinguishability implies ϵ/l-predictability.

Hybrid argument: if a distinguisher can ϵ-distinguish extreme hybrids given by p_f and p_{u_l}, then it can also distinguish adjacent hybrids given by q_{i-1} and q_i, with gap at least ϵ/l.

The distinguisher is used to produce a predictor.

The contrapositive is that unpredictability implies indistinguishability.