Cryptography Part III: Indistinguishability

ECE 4156/6156 Hardware-Oriented Security and Trust

Spring 2024

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

Reading Assignment

- Please read Chapter 3 of the course textbook by Katz and Lindell
- Please read Chapter 2 of the course textbook by Menezes, Oorschot and Vanstone, i.e., the Handbook of Applied Cryptography
    - Note: this book will be referred to later in these notes as “HAC”

Notation from HAC (page 49)

$Z^n = \{0,1,\ldots,n-$

- $\mathbb{R}$ is the set of real numbers
  - e.g., $\pi = \frac{\text{circum}}{\text{diam}}$
  - $\sqrt{-1}$ is **not** a real number but is imaginary, i.e., $\sqrt{-1} \notin \mathbb{R}$
- $\mathbb{Z}$ is the set of integers, i.e., $\mathbb{Z}$ = {…,-3,-2,-1,0,1,2,3,…}
- $f: A \rightarrow B$ is a function that maps each $a \in A$ to precisely one $b \in B$. Given that $f(a) = b$, then b is called the *image* of a, and a is called the *preimage* of b.

Additional Notation (~~from Prof. Mooney~~)

- $\mathbb{N}$ is the set of natural numbers, i.e., $\mathbb{N}$ = {1,2,3,…}

Notation from Katz and Lindell

- $\{X\}$ is a set of elements of type X
- m is a message in plaintext
  - m is composed of smaller blocks m_i suitable for individual encryption steps
  - $m = \{m_i\}$
- c_i is ciphertext corresponding to message block m_i
- c is ciphertext corresponding to message m
- Enc_k is encryption with key k
  - $c \leftarrow Enc_k(m)$
- Dec_k is decryption with key k
  - $m \leftarrow Dec_k(c)$
- $<a,b>$ is a concatenation of a followed by b

Negligible success probability. A negligible function is one that is asymptotically smaller than any inverse polynomial function. Formally:

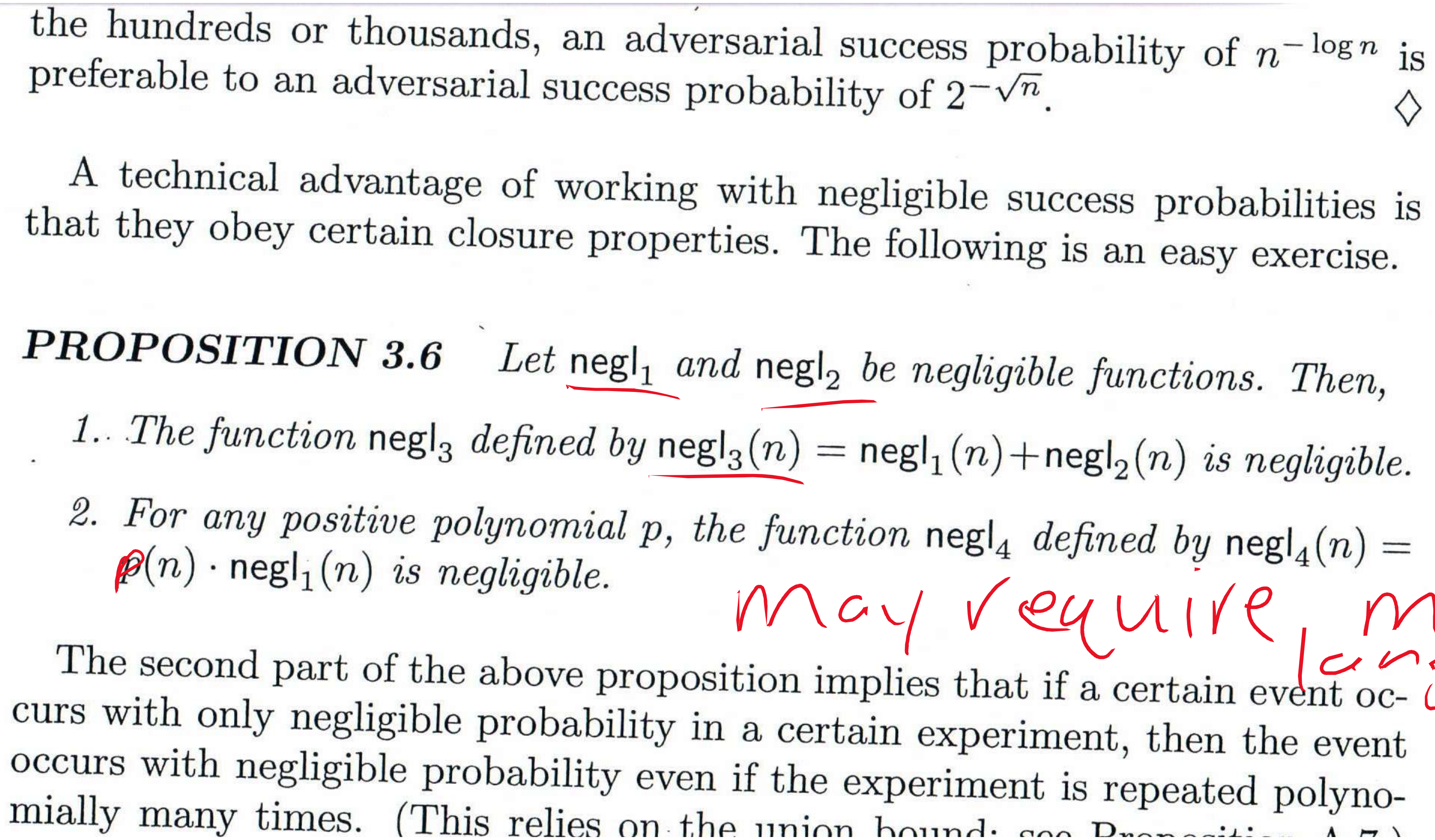
$f: \mathbb{N} \to \mathbb{R}$ $\quad p(n) = n^3$

DEFINITION 3.4 *A function f from the natural numbers to the non-negative real numbers is* negligible *if for every positive polynomial p there is an N* spec. *such that for all integers $n > N$* spec. *it holds that $f(n) < \frac{1}{p(n)}$.* $\quad \frac{1}{n^3}$

For shorthand, the above is also stated as follows: for every polynomial p and *all sufficiently large values of n* it holds that $f(n) < \frac{1}{p(n)}$. An equivalent formulation of the above is to require that for all constants c there exists an N such that for all $n > N$ it holds that $\boxed{f(n) < n^{-c}}$. We typically denote an arbitrary negligible function by $\boxed{\mathsf{negl}}$.

Example 3.5

The functions $\boxed{2^{-n}}$, $2^{-\sqrt{n}}$, and $\boxed{n^{-\log n}}$ are all negligible. However, they ap-



the hundreds or thousands, an adversarial success probability of $n^{-\log n}$ is preferable to an adversarial success probability of $2^{-\sqrt{n}}$. ◇

A technical advantage of working with negligible success probabilities is that they obey certain closure properties. The following is an easy exercise.

PROPOSITION 3.6 *Let* negl_1 *and* negl_2 *be negligible functions. Then,*

1. *The function* negl_3 *defined by* $\mathsf{negl}_3(n) = \mathsf{negl}_1(n) + \mathsf{negl}_2(n)$ *is negligible.*
2. *For any positive polynomial* p*, the function* negl_4 *defined by* $\mathsf{negl}_4(n) = p(n) \cdot \mathsf{negl}_1(n)$ *is negligible.*

may require much longer N pe

The second part of the above proposition implies that if a certain event occurs with only negligible probability in a certain experiment, then the event occurs with negligible probability even if the experiment is repeated polynomially many times. (This relies on the union bound; see Proposition A.7.)

symmetric

PPT

DEFINITION 3.7 *A* private-key encryption scheme *is a tuple of probabilistic polynomial-time algorithms* (Gen, Enc, Dec) *such that:*

1. *The* key-generation algorithm Gen *takes as input* 1^n *(i.e., the security parameter written in unary) and outputs a key* k*; we write* $k \leftarrow \mathsf{Gen}(1^n)$ *(emphasizing that* Gen *is a randomized algorithm). We assume without loss of generality that any key* k *output by* $\mathsf{Gen}(1^n)$ *satisfies* $|k| \geq n$.

2. *The* encryption algorithm Enc *takes as input a key* k *and a plaintext message* $m \in \{0,1\}^*$*, and outputs a ciphertext* c*. Since* Enc *may be randomized, we write this as* $c \leftarrow \mathsf{Enc}_k(m)$.

I W

ℓ(n)

3. *The* decryption algorithm Dec *takes as input a key* k *and a ciphertext* c*, and outputs a message* m *or an error. We assume that* Dec *is deterministic, and so write* $m := \mathsf{Dec}_k(c)$ *(assuming here that* Dec *does not return an error). We denote a generic error by the symbol* $\perp$.

⊥

n

It is required that for every n*, every key* k *output by* $\mathsf{Gen}(1^n)$*, and every* $m \in \{0,1\}^*$*, it holds that* $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$.

Bijective

If (Gen, Enc, Dec) *is such that for* k *output by* $\mathsf{Gen}(1^n)$*, algorithm* Enc_k *is only defined for messages* $m \in \{0,1\}^{\ell(n)}$*, then we say that* (Gen, Enc, Dec) *is a* fixed-length private-key encryption scheme for messages of length $\ell(n)$.

Almost always, $\mathsf{Gen}(1^n)$ simply outputs a uniform n-bit string as the key. When this is the case, we will omit Gen and simply define a private-key encryption scheme by a *pair* of algorithms (Enc, Dec).

Notation

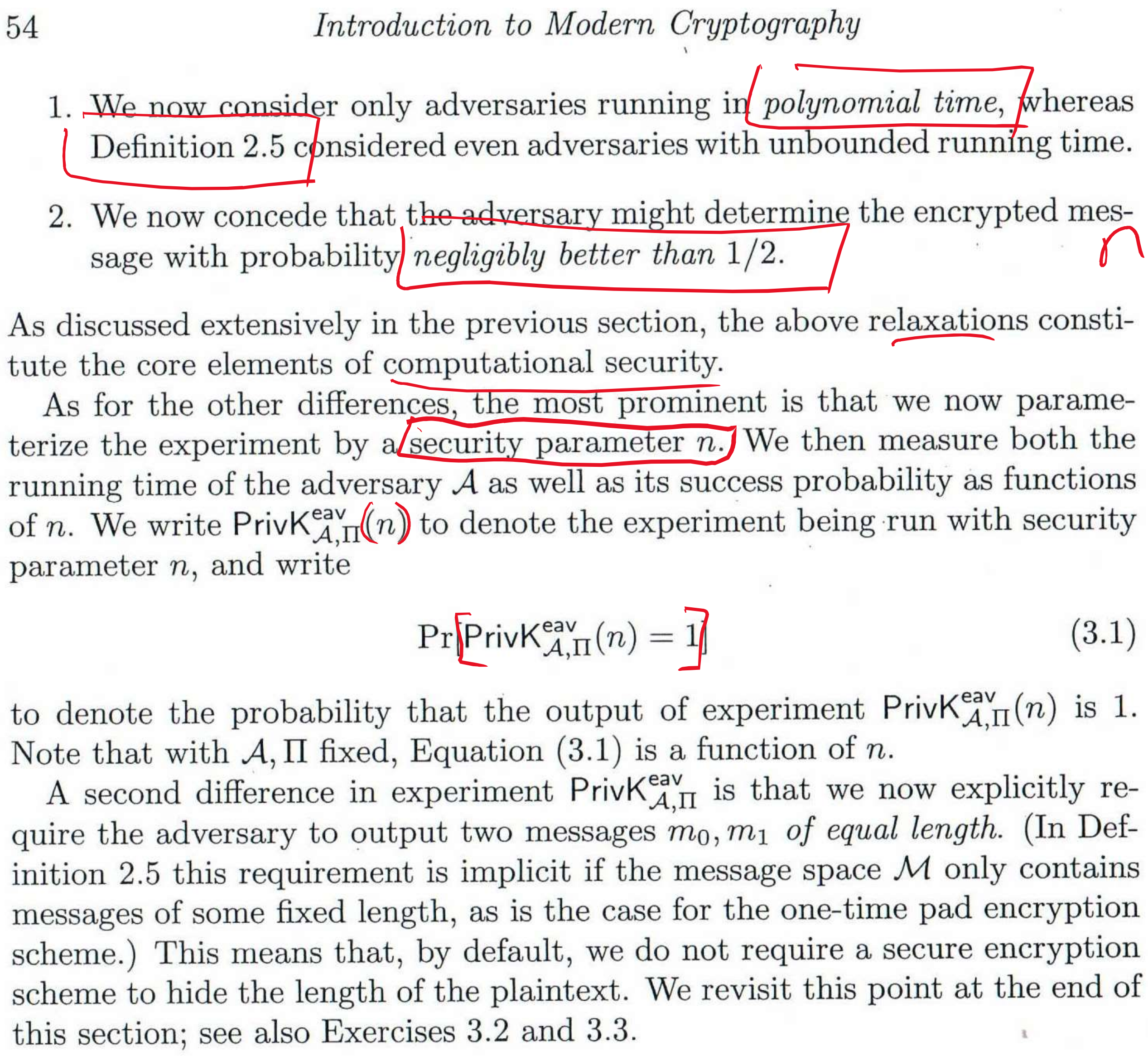
- PrivK is an experiment involving a private key
- A is an adversary
- eav refers to eavesdropping and obtaining ciphertext only
- π = (*Gen*, *Enc*, *Dec*) is an encryption scheme
- $\mathrm{PrivK}_{A,\pi}^{\mathrm{eav}}$ is an experiment involving a private key encryption scheme π with an adversary A only with access to ciphertext
- A does not have access to additional information, e.g., A does not have valid plaintext-ciphertext pairs obtained through other means

$n=56$

$\boxed{n=80}$

2^n

$n=128$



1. We now consider only adversaries running in *polynomial time,* whereas Definition 2.5 considered even adversaries with unbounded running time.

2. We now concede that the adversary might determine the encrypted message with probability *negligibly better than* 1/2.

As discussed extensively in the previous section, the above relaxations constitute the core elements of computational security.

As for the other differences, the most prominent is that we now parameterize the experiment by a security parameter n. We then measure both the running time of the adversary $\mathcal{A}$ as well as its success probability as functions of n. We write $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$ to denote the experiment being run with security parameter n, and write

$$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1] \tag{3.1}$$

to denote the probability that the output of experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$ is 1. Note that with $\mathcal{A}, \Pi$ fixed, Equation (3.1) is a function of n.

A second difference in experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}$ is that we now explicitly require the adversary to output two messages m_0, m_1 *of equal length.* (In Definition 2.5 this requirement is implicit if the message space $\mathcal{M}$ only contains messages of some fixed length, as is the case for the one-time pad encryption scheme.) This means that, by default, we do not require a secure encryption scheme to hide the length of the plaintext. We revisit this point at the end of this section; see also Exercises 3.2 and 3.3.

n^3

$\mathrm{negl}(n)+\frac{1}{2}$

$\frac{n^3}{2^n}$

Notation

- 1^n denotes 1 repeated *n* times, e.g., for *n* = 5 then we have that $1^n = 11111$
  - Note that in Professor Mooney's opinion sometimes Katz and Lindell use 1^n when *n* would have been just as clear (or even more clear!)
- b is a bit, i.e., it is possible that $b = 1$ or $b = 0$
- b' is a bit, i.e., it is possible that $b' = 1$ or $b' = 0$
  - Note that in Katz and Lindell the apostrophe $'$ does not signify complementation!
  - In other words, b' is just another variable such as $\tilde{b}$
  - As a result, it is possible to have both $b = 1$ and $b' = 1$
  - It is also possible to have both $b = 0$ and $b' = 0$

Indistinguishability in the presence of an eavesdropper. We now give the formal definition, beginning with the experiment outlined above. The experiment is defined for any private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, any adversary $\mathcal{A}$, and any value n for the security parameter:

Adv.

Server

The adversarial indistinguishability experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$:

n

K

m_0, m_1

1. *The adversary $\mathcal{A}$ is given input 1^n, and outputs a pair of messages m_0, m_1 with $|m_0| = |m_1|$.*

either $b=0$ or $b=1$

c = challenge ciphertext

2. *A key k is generated by running $\mathsf{Gen}(1^n)$, and a uniform bit $b \in \{0,1\}$ is chosen. Ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$. We refer to c as the* challenge ciphertext.

$c \leftarrow \mathsf{Enc}_k(m_b)$

poly time

either $b' = 0$ or $b' = 1$

3. *$\mathcal{A}$ outputs a bit $\tilde{b}'$.*
4. *The output of the experiment is defined to be $\boxed{1}$ if $\boxed{\tilde{b}' = b,}$ and 0 otherwise. If $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1$, we say that $\mathcal{A}$* succeeds.

There is no limitation on the lengths of m_0 and m_1, as long as they are the same. (Of course, if $\mathcal{A}$ runs in polynomial time, then m_0 and m_1 have length polynomial in n.) If Π is a fixed-length scheme for messages of length $\boxed{\ell(n)}$, the above experiment is modified by requiring $m_0, m_1 \in \{0,1\}^{\ell(n)}$.

Notation

- EAV-secure refers to secure against ciphertext only attacks
- Ciphertext only attacks are generally considered the lowest level of attack
  - More sophisticated attacks involve some amount of plaintext-ciphertext pairs
- Probabilistic Polynomial Time or PPT refers to algorithms which take at most polynomial time while having free use of a true random number generator
  - We will not cover the details in this course, but in general algorithms which have the capability of periodically making truly random choices are more powerful than algorithms which do not
  - For an example of polynomial time, consider a key of size n = 56 bits: an algorithm might take n^2 seconds to compute which is 56^2 s = 3,136 s = 52.3 minutes
  - On the other hand, 2^{56} s = 824 billion years

n large

DEFINITION 3.8 *A private-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable encryptions in the presence of an eavesdropper, *or is* EAV-secure, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$ *there is a negligible function* negl *such that, for all* n,

|e5a|

m_0, m_1

1^n

$\to 128$

$\to 192$

$\to 256$

3DES

$$\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \boxed{\mathsf{negl}(n)},$$

where the probability is taken over the randomness used by $\mathcal{A}$ *and the randomness used in the experiment (for choosing the key and the bit* b*, as well as any randomness used by* Enc*).*

Note: unless otherwise qualified, when we write "$f(n) \leq g(n)$" we mean that inequality holds for all n.

It should be clear that Definition 3.8 is *weaker* than Definition 2.5, which is equivalent to perfect secrecy. Thus, any perfectly secret encryption scheme has indistinguishable encryptions in the presence of an eavesdropper. Our goal, therefore, will be to show that there exist encryption schemes satisfying the above in which the key is shorter than the message. That is, we will show schemes that satisfy Definition 3.8 but cannot satisfy Definition 2.5.

= + 2

An equivalent formulation. Definition 3.8 requires that no PPT adversary can determine which of two messages was encrypted, with probability significantly better than 1/2. An equivalent formulation is that every PPT adversary *behaves the same* whether it sees an encryption of m_0 or of m_1. Since $\mathcal{A}$ outputs a single bit, "behaving the same" means it outputs 1 with almost the same probability in each case. To formalize this, define $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,b)$ as above except that the fixed bit b is used (rather than being chosen at random). Let $\mathsf{out}_{\mathcal{A}}(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,b))$ denote the output bit b' of $\mathcal{A}$ in the experiment. The following essentially states that no $\mathcal{A}$ can determine whether it is running in experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)$ or experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)$.

DEFINITION 3.9 *A private-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable encryptions in the presence of an eavesdropper *if for all* PPT *adversaries* $\mathcal{A}$ *there is a negligible function* negl *such that*

$$\left|\Pr[\mathsf{out}_{\mathcal{A}}(\underset{m_0}{\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)}) = 1] - \Pr[\mathsf{out}_{\mathcal{A}}(\underset{m_1}{\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)}) = 1]\right| \leq \mathsf{negl}(n).$$

The fact that this is equivalent to Definition 3.8 is left as an exercise.

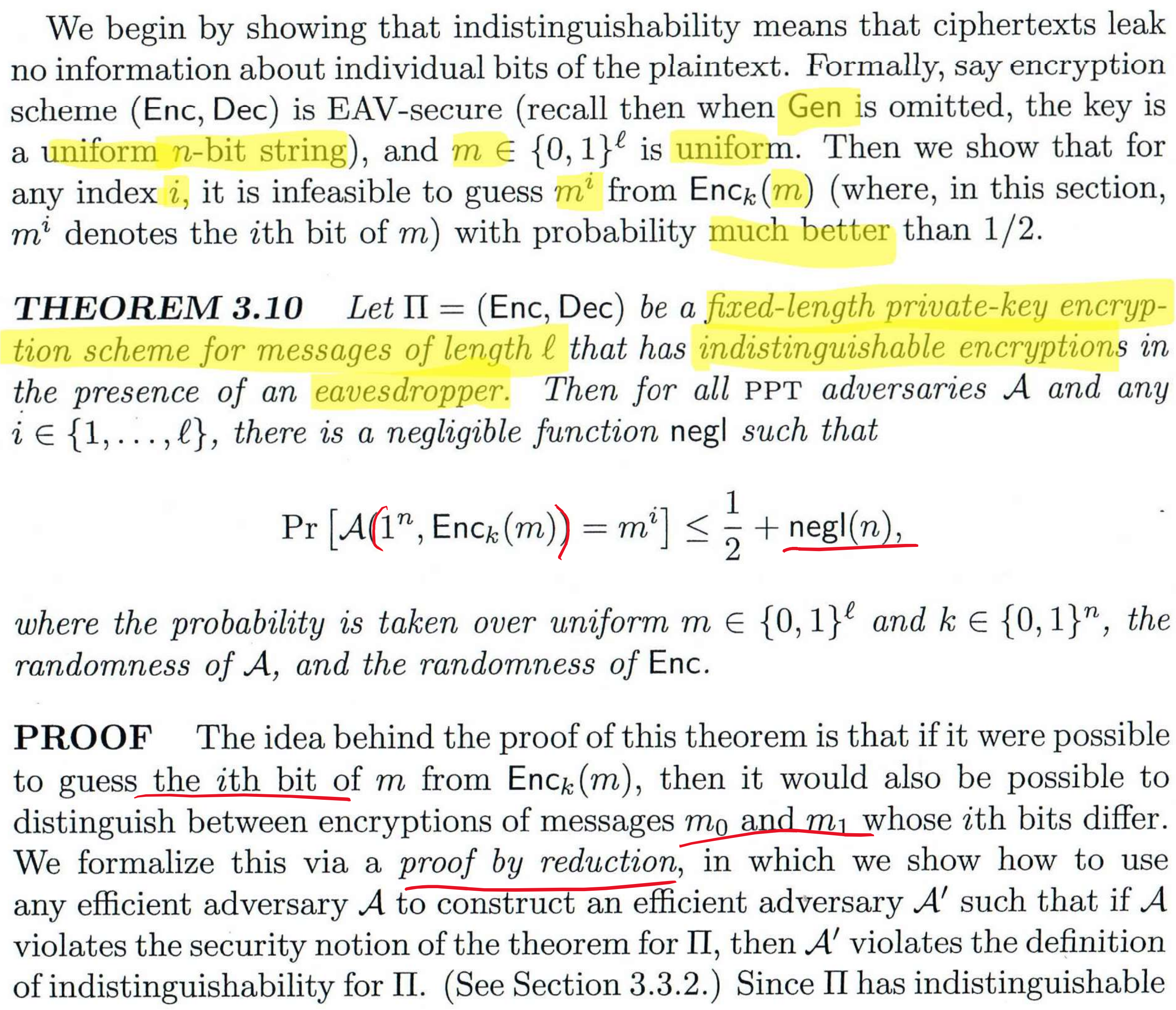
Notation in Theorem 3.10

- The plaintext message *m* has length ℓ
- *m* is a message in plaintext
  - *m* is composed of ℓ individual bits m^i , $1 \leq i \leq \ell$
  - note that elsewhere in the Katz and Lindell textbook m_i is a message block (i.e., multiple bits in a block)!

m^i

e.g., block size 128

$|m_i| = 128$



We begin by showing that indistinguishability means that ciphertexts leak no information about individual bits of the plaintext. Formally, say encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is EAV-secure (recall then when Gen is omitted, the key is a uniform n-bit string), and $m \in \{0,1\}^\ell$ is uniform. Then we show that for any index i, it is infeasible to guess m^i from $\mathsf{Enc}_k(m)$ (where, in this section, m^i denotes the ith bit of m) with probability much better than $1/2$.

THEOREM 3.10 *Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be a fixed-length private-key encryption scheme for messages of length ℓ that has indistinguishable encryptions in the presence of an eavesdropper. Then for all* PPT *adversaries $\mathcal{A}$ and any $i \in \{1, \ldots, \ell\}$, there is a negligible function* negl *such that*

$$\Pr\left[\mathcal{A}(1^n, \mathsf{Enc}_k(m)) = m^i\right] \le \frac{1}{2} + \mathsf{negl}(n),$$

where the probability is taken over uniform $m \in \{0,1\}^\ell$ and $k \in \{0,1\}^n$, the randomness of $\mathcal{A}$, and the randomness of Enc.

PROOF The idea behind the proof of this theorem is that if it were possible to guess the ith bit of m from $\mathsf{Enc}_k(m)$, then it would also be possible to distinguish between encryptions of messages m_0 and m_1 whose ith bits differ. We formalize this via a *proof by reduction*, in which we show how to use any efficient adversary $\mathcal{A}$ to construct an efficient adversary $\mathcal{A}'$ such that if $\mathcal{A}$ violates the security notion of the theorem for Π, then $\mathcal{A}'$ violates the definition of indistinguishability for Π. (See Section 3.3.2.) Since Π has indistinguishable

Notation and Example for Defn. 3.14

- *D* is a distinguisher which tests for randomness
- *G* is a candidate pseudorandom number generator which takes as input an *n*-bit seed *s* and outputs $\ell(n)$ random bits where $\ell(n) > n$
- Example where *D* can distinguish *G* from a truly random bit sequence
  - *G*(*s*) outputs seed *s* followed by the exclusive-or over all the seed bits, $\bigoplus_{i=1}^{n} s$
  - clearly $\ell(n) = n + 1$
  - Algorithm for *D*: given a “pseudorandom” sequence *w* from *G*, output a 1 iff the last bit of *w* equals the exclusive-or of all of the preceding bits of *w*
  - clearly *D* runs in polynomial time or less
  - $\Pr[D(G(s)) = 1] = 1$

is $\Pr[D(r)=1] = 0$? No, bec. r could choose for last bit true

DEFINITION 3.14 *Let ℓ be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n and any input $s \in \{0,1\}^n$, the result $G(s)$ is a string of length $\ell(n)$. We say that G is a* pseudorandom generator *if the following conditions hold:*

1. **(Expansion:)** *For every n it holds that $\ell(n) > n$.*
2. **(Pseudorandomness:)** *For any PPT algorithm D, there is a negligible function* negl *such that*

$$\left| \Pr[D(G(s)) = 1] - \Pr[D(r) = 1] \right| \leq \mathsf{negl}(n),$$

where the first probability is taken over uniform choice of $s \in \{0,1\}^n$ and the randomness of D, and the second probability is taken over uniform choice of $r \in \{0,1\}^{\ell(n)}$ and the randomness of D.

We call ℓ the expansion factor *of G.*

Notation in Theorem 3.10

$\ell(n) = GB$ $n=128$

- A stream cipher is traditionally defined as a variant of a pseudorandom number generator which generates random bits on demand
- Stream cipher G_ℓ maps an input of length n to an output of length $\ell(n) > n$

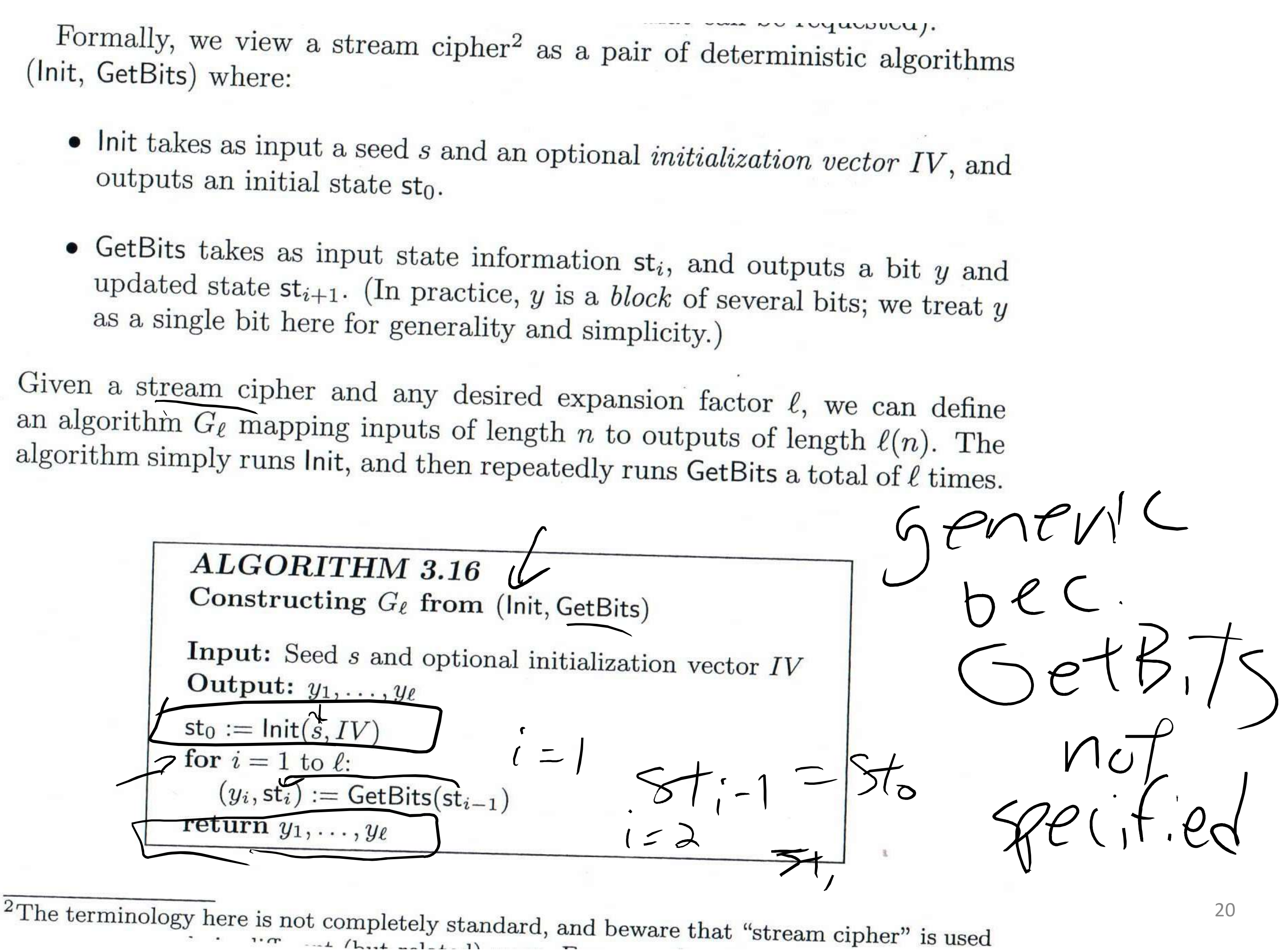
seed

- G_ℓ uses GetBits which takes input st_i and outputs y_{i+1}
- State information st_i is initially given value st_0 by Init which takes as input a seed s and an optional initialization vector IV

$IV_1 = 128$ bits $IV_2 = 128$ bits

$c \leftarrow Enc_k(m)$

$m := Dec_k(\quad)$



can be requested).

Formally, we view a stream cipher[2] as a pair of deterministic algorithms (Init, GetBits) where:

- Init takes as input a seed s and an optional *initialization vector IV*, and outputs an initial state st_0.
- GetBits takes as input state information st_i, and outputs a bit y and updated state st_{i+1}. (In practice, y is a *block* of several bits; we treat y as a single bit here for generality and simplicity.)

Given a stream cipher and any desired expansion factor ℓ, we can define an algorithm G_ℓ mapping inputs of length n to outputs of length $\ell(n)$. The algorithm simply runs Init, and then repeatedly runs GetBits a total of ℓ times.

ALGORITHM 3.16

Constructing G_ℓ from (Init, GetBits)

```
Input: Seed $s$ and optional initialization vector $IV$
Output: $y_1, \ldots, y_\ell$
$\mathsf{st}_0 := \mathsf{Init}(s, IV)$
for $i = 1$ to $\ell$:
    $(y_i, \mathsf{st}_i) := \mathsf{GetBits}(\mathsf{st}_{i-1})$
return $y_1, \ldots, y_\ell$
```

1

$i=1$

$st_{i-1} = st_0$

$i=2$

st_1

generic bec. GetBits not specified

[2] The terminology here is not completely standard, and beware that "stream cipher" is used

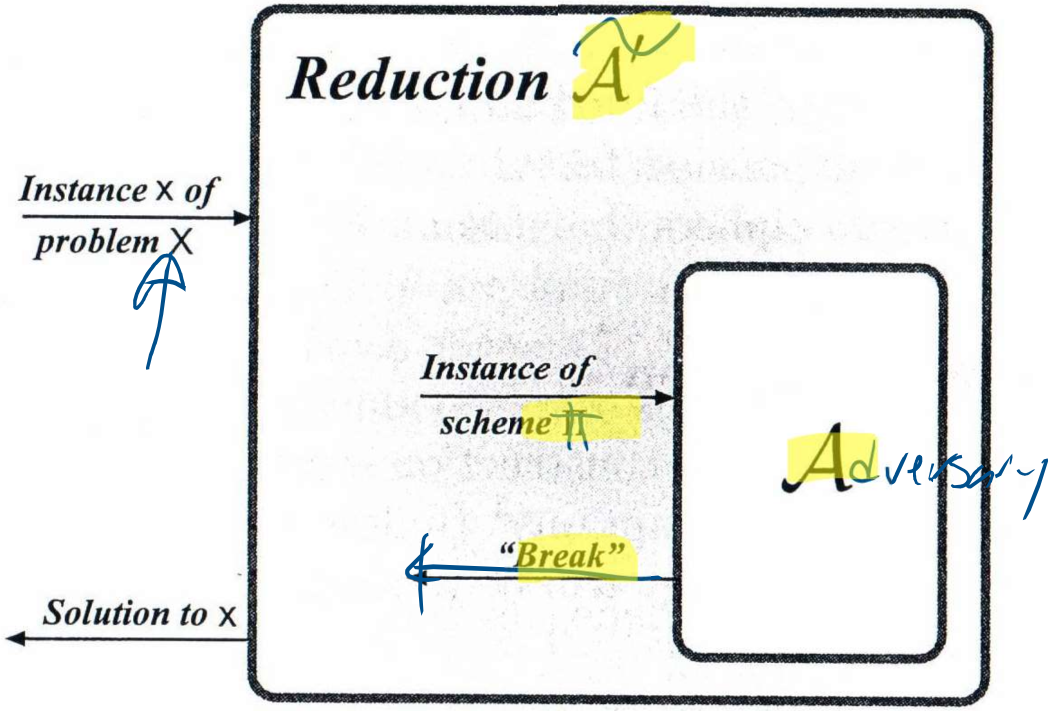
For ex., given $n = p \cdot q$ (2048 bits), p prime, q prime,

Proof by Reduction: Application to Crypto

no known polynomial time (or better) algorithm to find p or q

- PPT adversary A attacks encryption scheme π = (*Gen*, *Enc*, *Dec*)
  - A has non-negligible probability ε(*n*) of succeeding in breaking π
- Assume problem X cannot be solved by any PPT adversary
  - E.g., problem X could involve finding large prime factors of large numbers, or distinguishing a particular form of pseudorandom number generator from a true
- Construct A' which is called the reduction; A' uses A to attack X
- If A' succeeds, then A' can solve problem X in polynomial time which contradicts our best current knowledge / assumptions
- Therefore no PPT adversary A can successfully attack encryption scheme π with a non-negligible probability of success

66 *Introduction to Modern Cryptography*

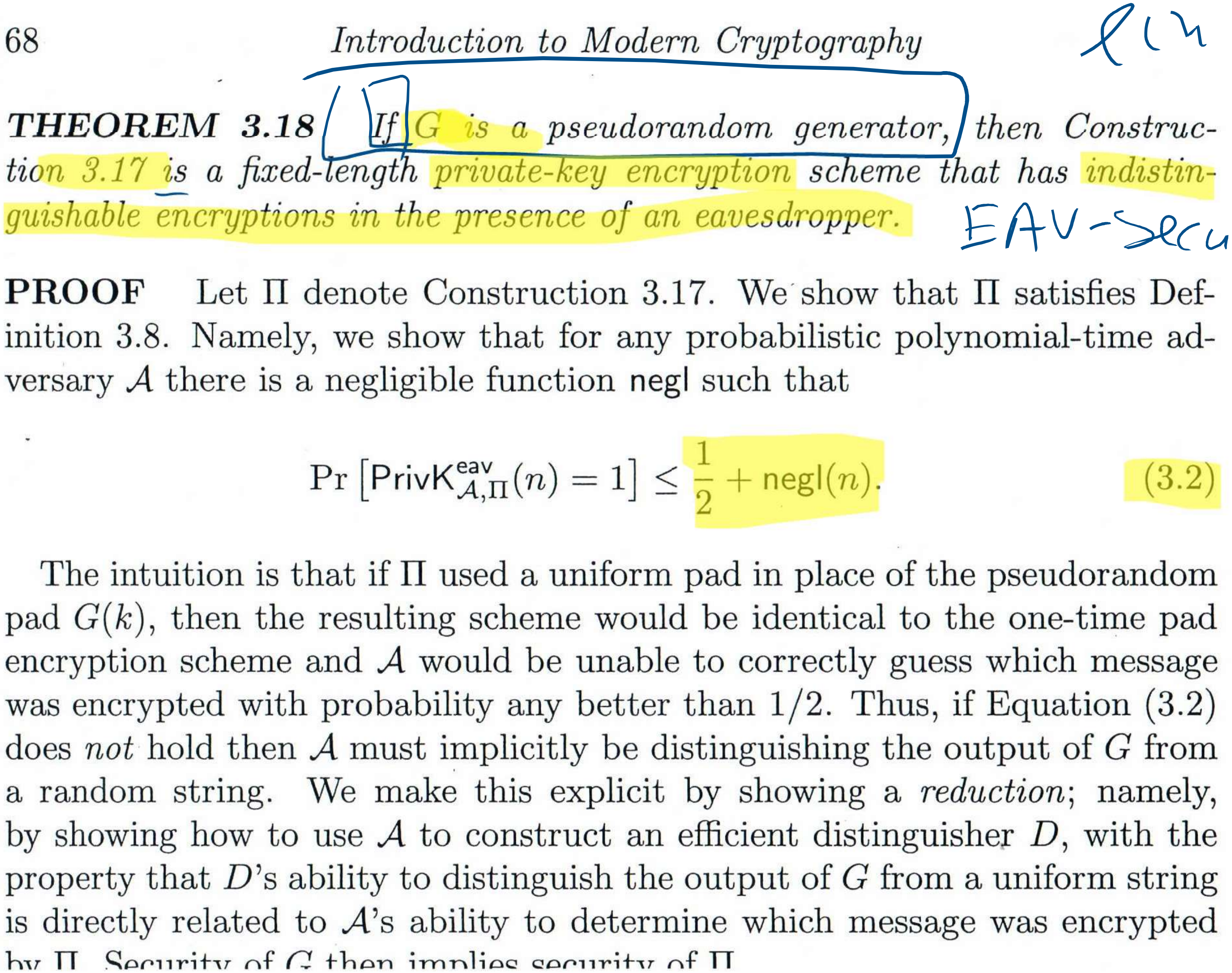


enc. scheme Π = (Gen, Enc, Dec)

FIGURE 3.1: A high-level overview of a security proof by reduction.

Comments

- Please note that we are still exclusively considering ciphertext-only attacks!
- In what follows we will consider a pseudorandom number generator G with an expansion factor $\ell(n)$ where $\ell(n) > n$



$\ell(n)$

THEOREM 3.18 *If G is a pseudorandom generator, then Construction 3.17 is a fixed-length private-key encryption scheme that has indistinguishable encryptions in the presence of an eavesdropper.*

10^6

EAV-Secure 10^{12}

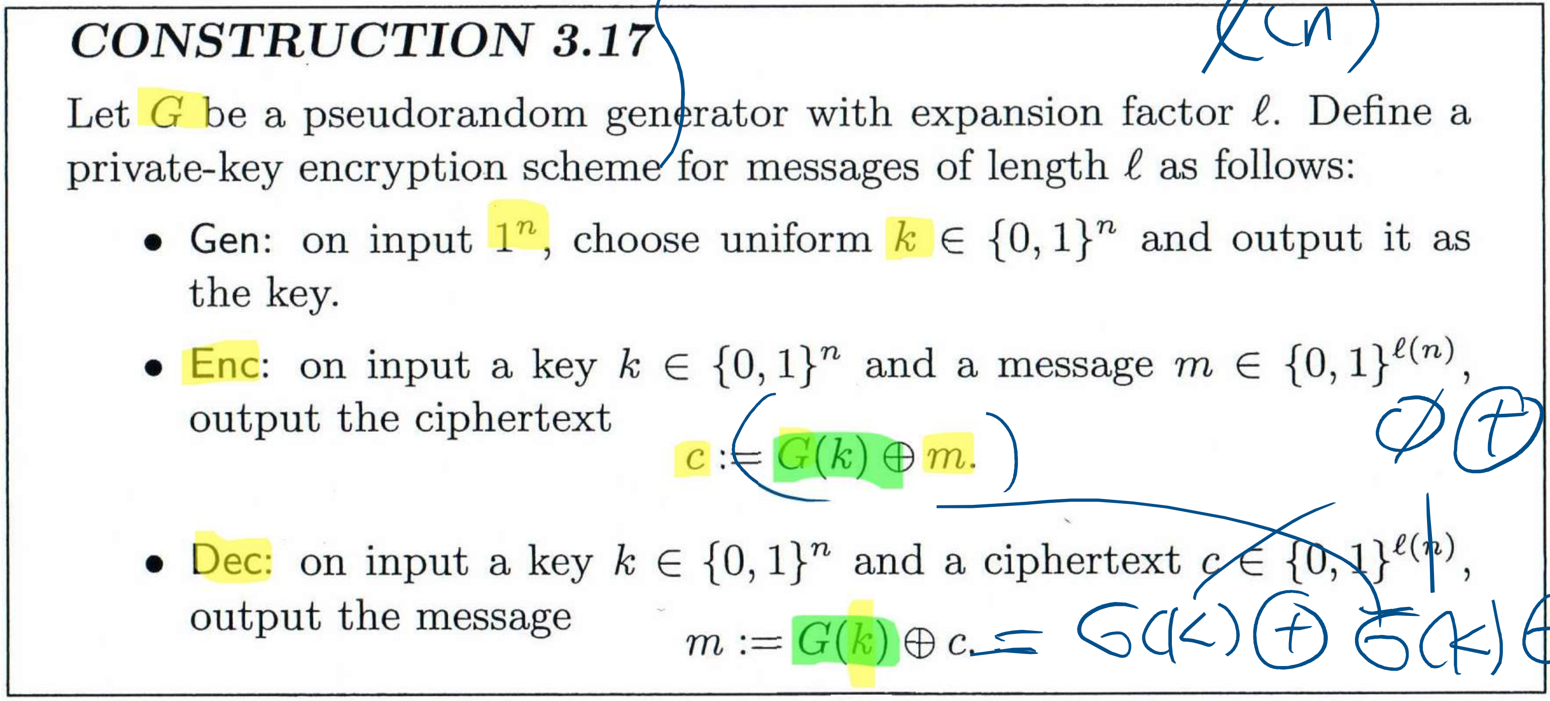
PROOF Let Π denote Construction 3.17. We show that Π satisfies Definition 3.8. Namely, we show that for any probabilistic polynomial-time adversary $\mathcal{A}$ there is a negligible function negl such that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)=1\right] \le \frac{1}{2}+\mathsf{negl}(n). \tag{3.2}$$

The intuition is that if Π used a uniform pad in place of the pseudorandom pad $G(k)$, then the resulting scheme would be identical to the one-time pad encryption scheme and $\mathcal{A}$ would be unable to correctly guess which message was encrypted with probability any better than $1/2$. Thus, if Equation (3.2) does *not* hold then $\mathcal{A}$ must implicitly be distinguishing the output of G from a random string. We make this explicit by showing a *reduction*; namely, by showing how to use $\mathcal{A}$ to construct an efficient distinguisher D, with the property that D's ability to distinguish the output of G from a uniform string is directly related to $\mathcal{A}$'s ability to determine which message was encrypted by Π. Security of G then implies security of Π.

1^n n bits

Π



CONSTRUCTION 3.17

$\ell(n)$

Let G be a pseudorandom generator with expansion factor ℓ. Define a private-key encryption scheme for messages of length ℓ as follows:

- Gen: on input 1^n, choose uniform $k \in \{0,1\}^n$ and output it as the key.
- Enc: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^{\ell(n)}$, output the ciphertext
$$c := G(k) \oplus m.$$
$0 \oplus m = m$
- Dec: on input a key $k \in \{0,1\}^n$ and a ciphertext $c \in \{0,1\}^{\ell(n)}$, output the message
$$m := G(k) \oplus c. = G(k) \oplus G(k) \oplus m$$

A private-key encryption scheme based on any pseudorandom generator.

Proof Sketch

if and only if

- Suppose G were to be replaced with a one-time pad: then no adversary would be able to solve Construction 3.17 with non-negligible probability
- Thus, an adversary A can overcome Eqn. (3.2) iff A can distinguish a pseudorandom number generator G from a true random number generator in polynomial time or less
  - Reduction A' is constructed to distinguish a pseudorandom number generator from a true random number generator
- However, we assume that we have chosen a pseudorandom number generator that cannot be distinguished from a true random number generator in polynomial time
- Therefore, no efficient adversary can solve Construction 3.17

A iff B A ⇒ B B ⇒ A

Notation for Concrete Security

$t = 2^{80}$ ns

- Assume n fixed, e.g., n = 128
- Let G run in time at most $\boxed{t}$
  - e.g., consider $t = 2^{80}$ nanoseconds = 13 trillion years
- Say that any distinguisher D should have a fixed probability of less than ε of succeeding
  - e.g., consider $ε = 2^{-60}$

prob. D succeeds is less than $\frac{1}{2^{60}}$

$n=80$

$\Rightarrow t = 2^{80}$

$2^{-60} = \varepsilon$

Concrete security. Although Theorem 3.18 and its proof are in an asymptotic setting, we can readily adapt the proof to bound the *concrete* security of the encryption scheme in terms of the concrete security of G. Fix some value of n for the remainder of this discussion, and let Π now denote Construction 3.17 using this value of n. Assume G is (t, ε)-pseudorandom (for the given value of n), in the sense that for all distinguishers D running in time at most t we have

G(S) is Pseudorandom

$$\left|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]\right| \leq \varepsilon. \tag{3.6}$$

(Think of $t \approx 2^{80}$ and $\varepsilon \approx 2^{-60}$, though precise values are irrelevant for our discussion.) We claim that Π is $(t - c, \varepsilon)$-secure for some (small) constant c, in the sense that for all $\mathcal{A}$ running in time at most $t - c$ we have

$$\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} = 1\right] \leq \frac{1}{2} + \varepsilon. \tag{3.7}$$

replace negl w/ ε

(Note that the above are now fixed numbers, not functions of n, since we are not in an asymptotic setting here.) To see this, let $\mathcal{A}$ be an arbitrary

Chosen Plaintext Attacks

- In the real world, an adversary may be able to obtain ciphertext corresponding to some chosen plaintexts, e.g., via a low level insider organizing initialization sequences each day
- CPA models can be analyzed using an *encryption oracle* which can encrypt plaintexts other than the ones under attack

A

m_0, m_1

poly sets

$\mathsf{Enc}_k(m_j) = c_j$

$j = 0, 1$

C

box" that encrypts messages of $\mathcal{A}$'s choice using a key k that is unknown to $\mathcal{A}$. That is, we imagine $\mathcal{A}$ has access to an "oracle" $\mathsf{Enc}_k(\cdot)$; when $\mathcal{A}$ *queries* this oracle by providing it with a message m as input, the oracle returns a ciphertext $c \leftarrow \mathsf{Enc}_k(m)$ as the reply. (When Enc is randomized, the oracle uses fresh randomness each time it answers a query.) The adversary is allowed to interact with the encryption oracle adaptively, as many times as it likes.

Consider the following experiment defined for any encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, adversary $\mathcal{A}$, and value n for the security parameter:

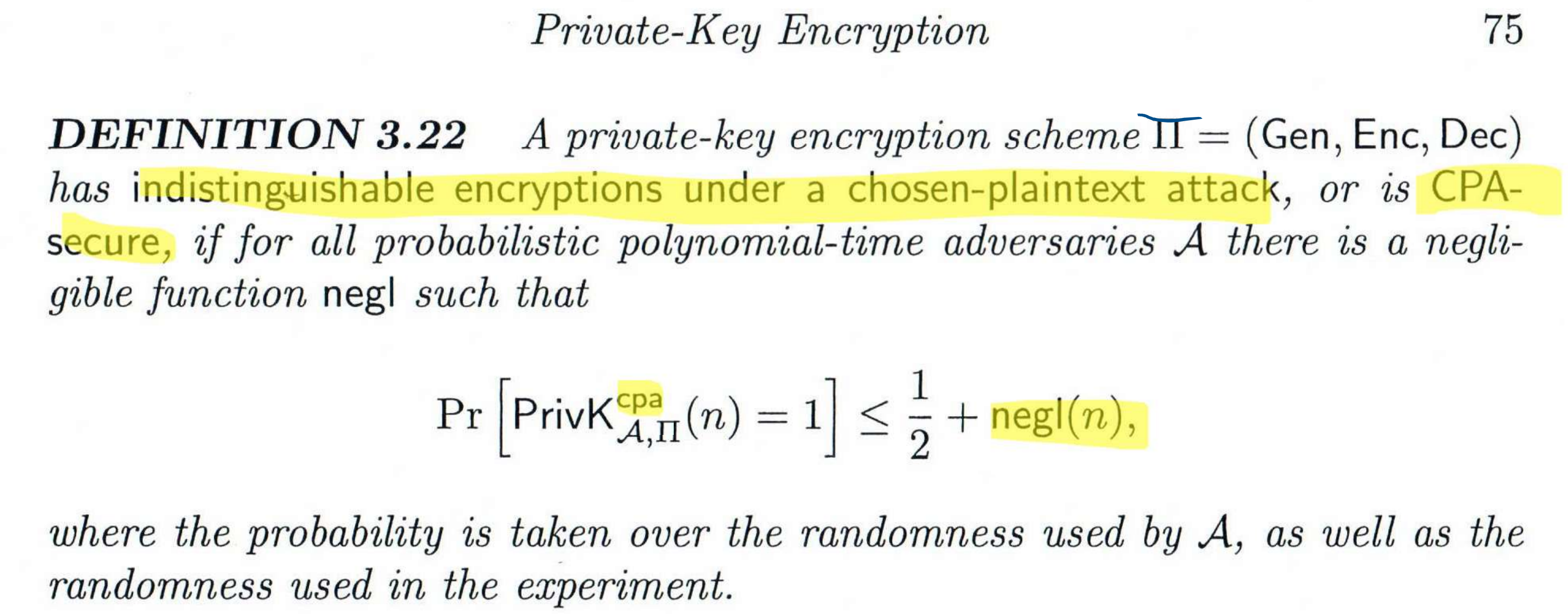
Server

K

b

The CPA indistinguishability experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$**:**

1. *A key k is generated by running $\mathsf{Gen}(1^n)$.*
2. *The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a pair of messages m_0, m_1 of the same length.*
3. *A uniform bit $b \in \{0,1\}$ is chosen, and then a ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$.*
4. *The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a bit b'.*
5. *The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. In the former case, we say that $\mathcal{A}$* succeeds.



Private-Key Encryption 75

DEFINITION 3.22 *A private-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable encryptions under a chosen-plaintext attack, *or is* CPA-secure, *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$ *there is a negligible function* negl *such that*

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n),$$

where the probability is taken over the randomness used by $\mathcal{A}$, *as well as the randomness used in the experiment.*