EECS 498/598: Encrypted Systems
Winter 2022
Lecture 2: Crypto Preliminaries

Paul Grubbs

paulgrub@umich.edu

Beyster 4709

Agenda for this lecture

- Announcements, introduction, notation
  - PRFs, concrete vs. asymptotic security
- Symmetric-key cryptography
  - PRPs, MACs, AEAD
  - The random oracle model
- Asymmetric cryptography
  - PKE, signatures, key exchange
- Discussion

Announcements

- Lecture schedule
- Canvas submission for paper reviews

Introduction and notation: PRFs

Reminder about applied crypto

Theory and practice

Syntax
Semantics
Security definitions

Instantiations
Use cases
performance (typical)

Introduction and notation: PRFs

Pseudo random function

$F: \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}$

$\mathcal{K}$: key space of PRF

$\mathcal{X}$: input space

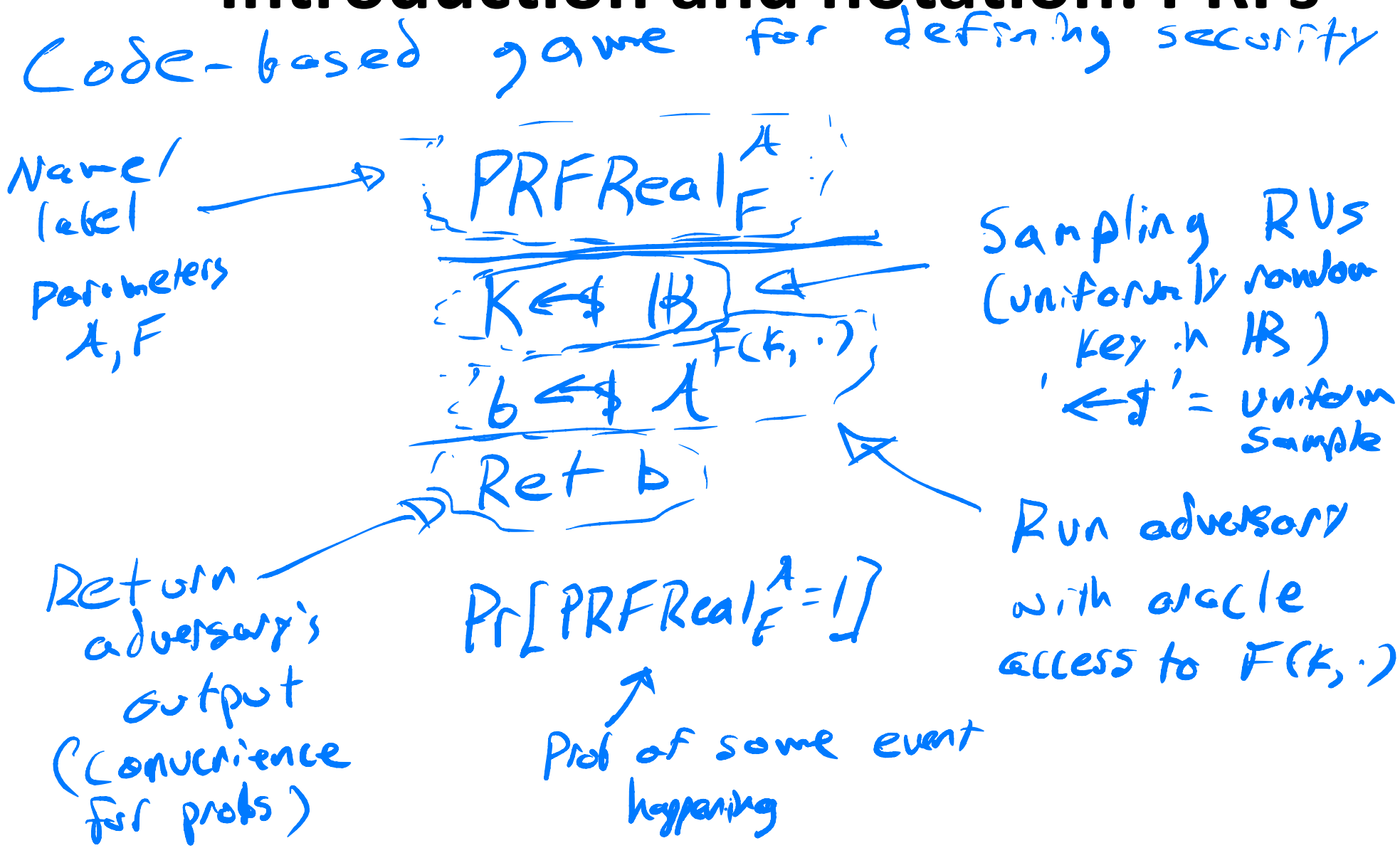
$\mathcal{Y}$: output space

Function that is "random-looking":

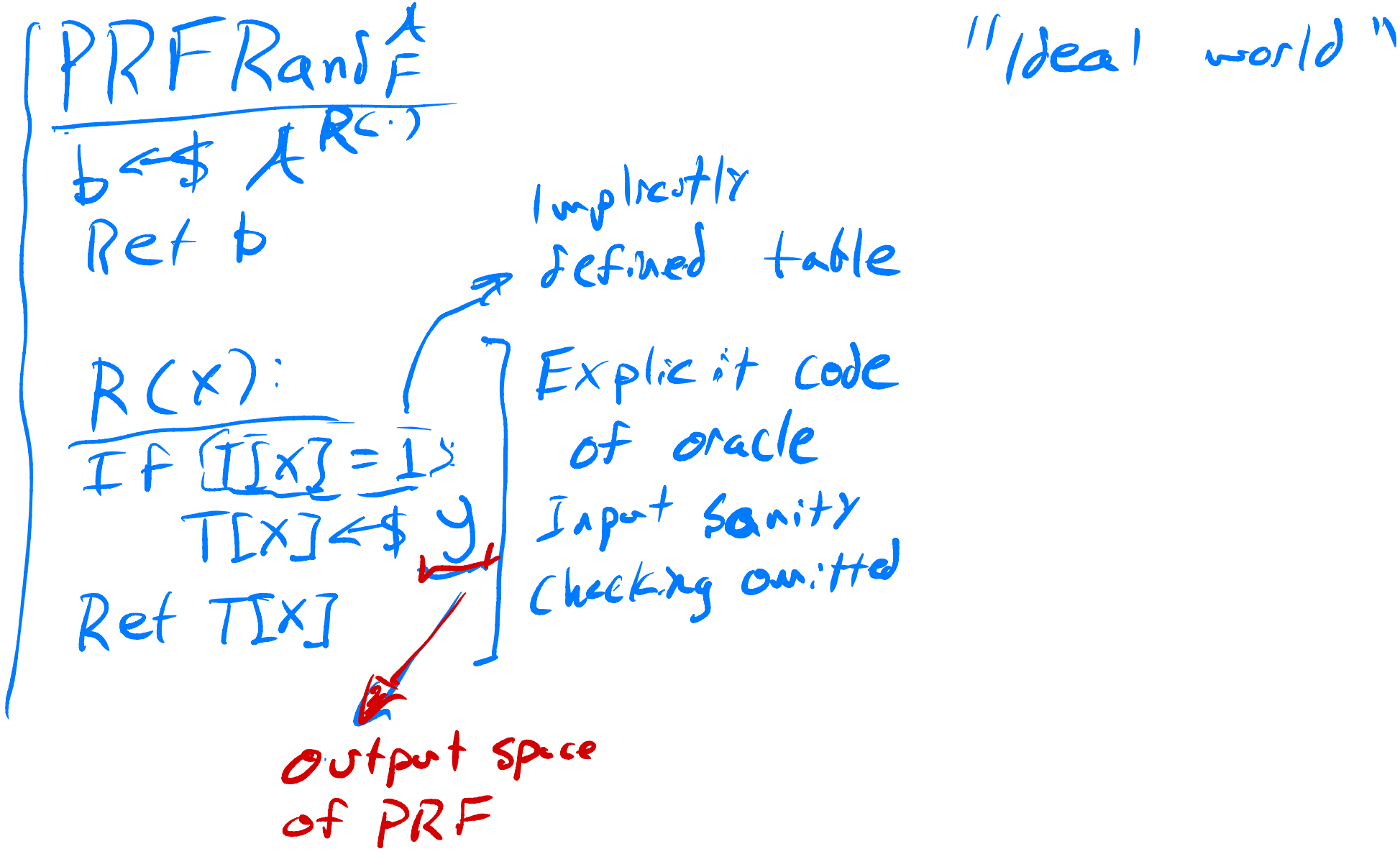
output looks like a random sample in $\mathcal{Y}$

→ Make this precise?

Introduction and notation: PRFs



Introduction and notation: PRFs



Concrete vs. asymptotic security

1. Concrete: measure advantage of adversary against fixed function (input/output space)

$$Adv_{PRF}(A, F) = \left| Pr[PRFReal_F^A = 1] - Pr[PRFRand_F^A = 1] \right|$$

2. Asymptotic: Define PRF security as function of 'security parameter' that grows to infinity.

F is a good PRF if for all nuppt adversaries,

$$Adv_{PRF}(A, F) = negl(\lambda)$$

Concrete vs. asymptotic security

Asymptotic:

+ Simplifies analysis
- Harder to reason about security for fixed F's
  E.g. AES only defined for 3 params!
- Need to parameterize everything by sec. param.

Concrete:

+ captures exact runtimes and advantages
+ security theorems lead to easy bit security estimates
- analyses much more complicated. Sensitive to model

- I'll use concrete security in lectures.
  You'll see both
- PRF instantiations
  - HMAC-SHA-256
  - Blake2
  - SHA3

Perf:

~ 1 gigabyte/second

Agenda for this lecture

- Announcements, introduction, notation
  - PRFs, concrete vs. asymptotic security
- Symmetric-key cryptography
  - PRPs, MACs, AEAD
  - The random oracle model
- Asymmetric cryptography
  - PKE, signatures, key exchange
- Discussion

Pseudorandom permutations (PRPs)

block cipher

$E : \mathcal{B} \times \mathcal{M} \to \mathcal{M}$

- permutation for all keys
- can invert given the key

$\underline{\text{Real}_E^{\mathcal{A}}}$

$K \xleftarrow{\$} \mathcal{K}$

$b \xleftarrow{\$} \mathcal{A}^{E(k,\cdot)}$

Ret b

$\underline{\text{Rand}_E^{\mathcal{A}}}$:

$\pi \xleftarrow{\$} \text{Perms}(\mathcal{M})$

$b \xleftarrow{\$} \mathcal{A}^{\pi(\cdot)}$

Ret b

$$\text{Adv}^{\text{PRP}}(\mathcal{A}, E) = \left| \Pr[\text{Real}_E^{\mathcal{A}} = 1] - \Pr[\text{Rand}_E^{\mathcal{A}} = 1] \right|$$

Pseudorandom permutations (PRPs)

Instantiations:

- AES
  - Amazing Design
  - Most CPUs have AES instruction
- 3DES / DES
  - First standard for commercial crypto

Use cases:

- Data encryption (building block of AEADs)
- pseudo random number generator
  - → /dev/urandom

Perf:

> 8 million calls per second

Very fast!

Message authentication codes (MACs)

$MAC = (Kg, Tag, Ver)$

- Kg: keygen
- Tag: Takes key/msg, outputs tag
- Ver: Takes key/msg and tag, outputs 0/1

$UFCMA^{A}_{MAC}$:

$K \leftarrow\$ Kg$

$(m,t) \leftarrow\$ A^{Tag(K,\cdot)}$

Ret $m \notin T \wedge Ver(K,m,t) = 1$

Rules out "trivial wins"!

$Tag(K,m)$:

$T \leftarrow T \cup \{m\}$

Ret $Tag(K,m)$

Measures A's ability to "forge" tag for new message

"UFCMA" game:

Identical except no m&T check.

$Pr[UFCMA^{A}_{MAC} = 1]$

$= 1$

$A_0^{Tag(\cdot)}$:

$t \leftarrow Tag("hello")$

Ret "hello", t

Message authentication codes (MACs)

Instantiations:

- HMAC-SHA-256
- common → Carter-Wegman MACs
  - GMAC, Poly1305
  - AES-CMAC

Use cases:

- secure channels
- cookies authentication

Performance:

comparable to PRF. Very fast

Authenticated encryption with associated data (AEAD)

AEAD = (Kg, Enc, Dec) → key, ciphertext, header

Kg: keygen

Enc: takes key, message, header

* combined confidentiality + integrity
* integrity-protected associated data (e.g. IP packet header)
* security:
  - ciphertexts "look like" random bits
  - can't create valid ciphertext without key

Authenticated encryption with associated data (AEAD)

Instantiations:

- AES-CBC-then-HMAC-SHA-256
- GCM, Chacha20/Poly1305

Use cases:

- encrypt data in network protocols

Perf:

AES-GCM: 12 million/second

(possible to get ~1 cycle/byte!)

Random Oracle Model

For time reasons, skip.

Resources on Piazza

Agenda for this lecture

- Announcements, introduction, notation
  - PRFs, concrete vs. asymptotic security
- Symmetric-key cryptography
  - PRPs, MACs, AEAD
  - The random oracle model
- Asymmetric cryptography
  - PKE, signatures, key exchange
- Discussion

Public-Key Encryption

PKE = (Kg, Enc, Dec) → private key and ciphertext

- Kg: outputs public key and private key
- Enc: takes public key and message

* encrypt data to someone w/o shared secret

Use cases:
- key establishment
- cryptocurrency (privacy in payments)

Perf:
- ~1000x slower than AEAD
- Enc for RSA: 17.6K/sec
  Dec for RSA: 900/sec

Instantiations:
- RSA encryption [RSA78]
- ElGamal and variants
  - $\mathbb{Z}_p^*$ for $p \simeq 2\text{-}3k$ bits
  - Now: Elliptic curves
  - Post-quantum crypto
    Chris Peikert
    Ask Chad!!

Digital Signatures

$SIG = (kg, sign, verify)$

- kg → pub key, priv key
- $sign$ (priv key, message) → outputs signature
- $verify$ (public key, signature, message) → outputs 0/1

* Integrity protection + authentication
* Security: unforgeability

Use cases:
- authenticate server in TLS
- Code signing in package manager
- transaction signing in crypto currencies

Instantiations:
- RSA sigs
- DSA
- ECDSA
- Schnorr
- BLS ← pairings of elliptic curves

Perf:
- 3.3K sigs/second
- 4.2K verify/second

Key Exchange

- Interactive protocol between ≥2 parties to agree on shared secret
- security models complex, out of scope
  - confidentiality of key: secret should
  - authenticity of parties

Use case:

Key establishment in TLS, IPSec, Signal...

Perf:

1.3*ECDHE/second

(How many does google do?)

Instantiations:

Diffie-Hellman [DH76]

today:

ECDHE — ephemeral (forward secrecy) Next week!

RSA etc

Agenda for this lecture

- Announcements, introduction, notation
  - PRFs, concrete vs. asymptotic security
- Symmetric-key cryptography
  - PRPs, MACs, AEAD
  - The random oracle model
- Asymmetric cryptography
  - PKE, signatures, key exchange
- Discussion

Discussion

- (Many) more topics will be covered this semester.
- Not very familiar with something we discussed?
  - Don’t panic!
- Learning things independently is a “meta”-skill. Takes practice