

lec01: Intro, PRGs

cs8803pc.cc.gatech.edu

Welcome! CS 8803 (14803) Programmable Crypto

Prof: Alex Ozdemir

New SCP Prof.

Research: full-stack crypto

crypto, applications, compilers, verification

Taught this @ Stanford: '21, '22, '23

Thrilled to teach it again.

New parts: programming, focus on programmable crypto

Today:

[1] What course is about

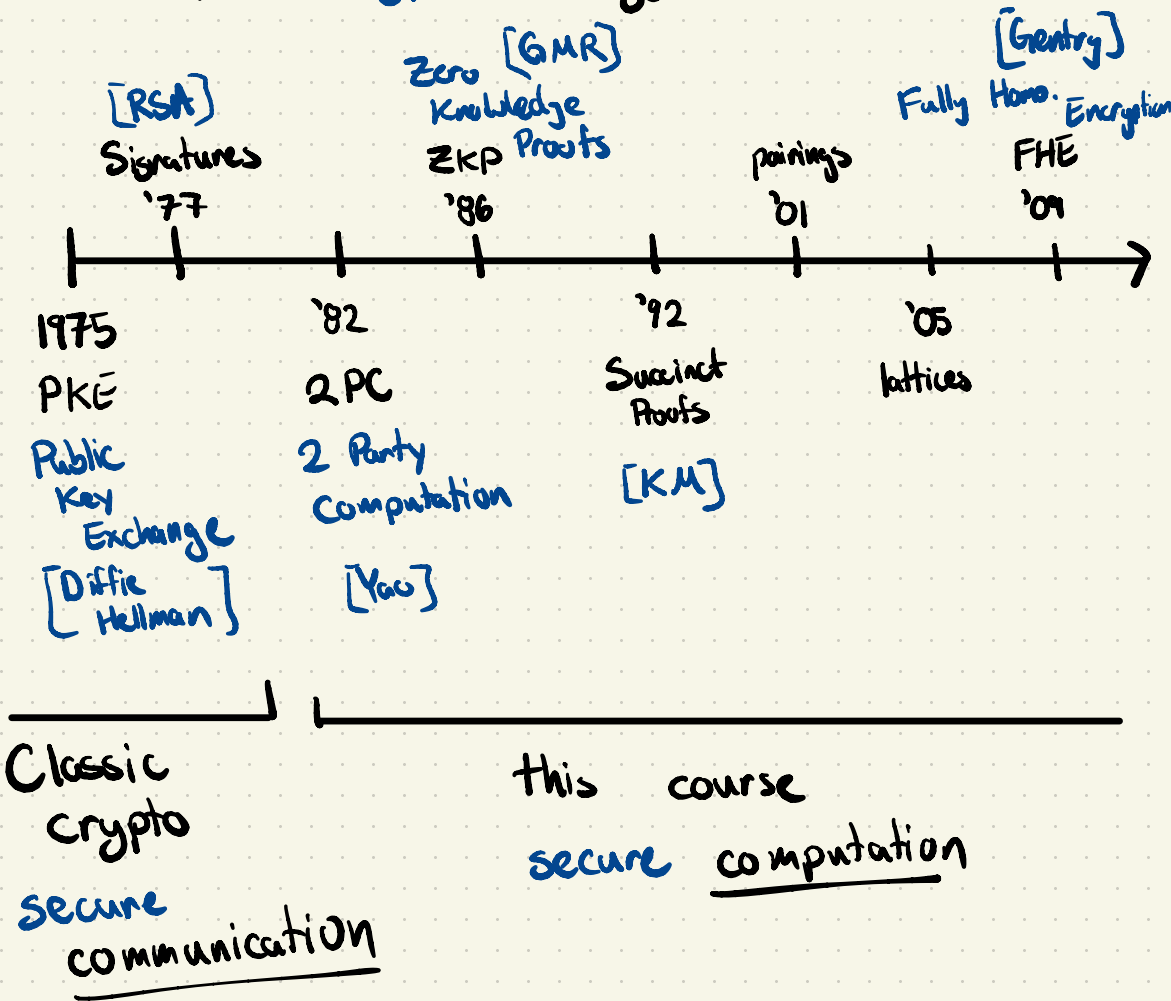
- Physical demo

[2] Logistics

[3] PseudoRandom Generators (PRGs)

I Overview

History of crypto (theory)



Demo:

- Where's Waldo?

Goal: Prove Waldo is there
without showing where

input: location

output: presence

- Romantic cryptography

Goal: discover mutual interest w/o embarrassment

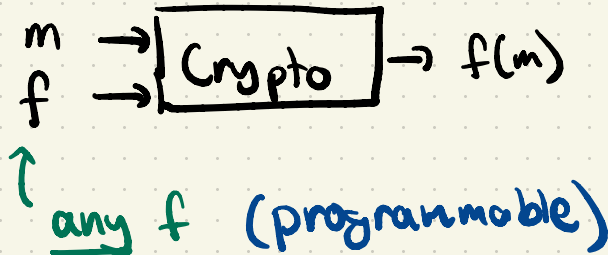
input: a, b

output: $\text{AND}(a, b)$

- This course:

- Digital tools (not physical)

- programmable tools



• Topics:

- 1) Foundations
- 2) Zero Knowledge Proofs
- 3) Succinct Proofs
- 4) Multiparty Computation
- 5) Fully Homomorphic Encryption

} different
Programmable
Cryptosystems

• Skills:

- What PCs do
- How they work ← math
- How to use/implement them ← programming

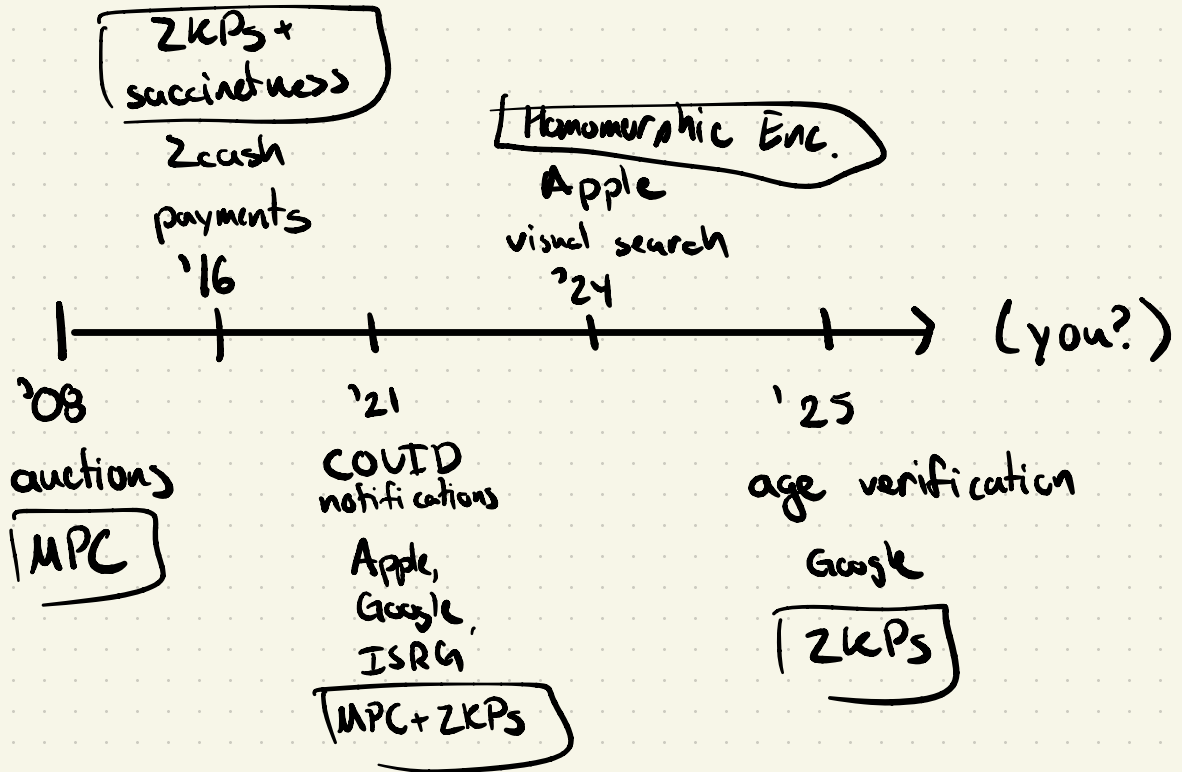
Course pitch:

- beautiful math
- about privacy, power, ...
- deployed

Anti-pitch:

- required background:
 - math + proofs (logic, not fancy math)
 - programming (Rust)
- time-consuming

Deployment history



2 Logistics

Website: cs8803pc.cc.gatech.edu

Contact : - Ed

- Anonymous feedback form
- Gradescope
- email

Lectures : • notes

- no recording
- no textbook (ref: cryptobook.us)
- attendance encouraged

Homework : - every 2 weeks (60%)

- due: Fri. @ 3pm (1st: next week)
- covers that Mon.

- written + programming

- work together

- write solo

- no Google or AI

- 4 late days (no partials)

Office Hours : - group work environment

- email for 1-1

Midterm Exam (15%)

Final Options: (25%)

- Exam, or
- Project! (starts after midterm)
 - Implementation
 - Theory (Reading / Research)

3 PRG

(First,)
Basics

Crypto is about hardness

- honest parties are efficient: $\text{poly}(\lambda)$ ← tunable
- best attacks are not

Ex: Encryption:

- with key, Enc/Dec $\text{poly}(\lambda)$
- w/o key, infeasible

Security holds under assumptions

↑ known hard problems
e.g. factoring

Security proof

hard problem $m \rightarrow$ secure system

factoring is hard $\rightarrow$ signature is unforgeable

Done by "reduction": contrapositive

attack on problem $\leftarrow$ attack on system

factoring alg. $\leftarrow$ forgery algorithm

Crypto: reduction graph

$\text{OWF} \Rightarrow \text{PRG} \Rightarrow \text{PRF} \Rightarrow \text{Block Ciphers}$

What are these basic assumptions?

- factoring, d-log, DDH, CDH, AES is a PRP, ...

Today + Wed:

Pseudo Random Generators

PRGs matter because they:

1. give practical encryption (ChaCha, in TLS, SSH, phones, ...)
2. foundational in symmetric crypto:

OWF $\Rightarrow$ PRG $\Rightarrow$ PRF $\Rightarrow$ BlockCipher $\Rightarrow$ Auth. Enc.

3. beautifully illustrate indistinguishability

$\Uparrow$ key to ZKP, MPC, ...

Syntax:

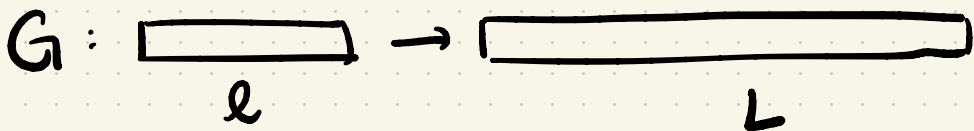
A PRG is an efficient, deterministic alg:

$$\begin{array}{ccccc} G : S & \rightarrow & R \\ \uparrow & \uparrow & \uparrow \\ \text{PRG} & \text{seed space} & \text{"random" space} \end{array}$$

typically:

$$S = \{0,1\}^L \leftarrow \text{length-}L \text{ bitstrings}$$

$$R = \{0,1\}^L \quad L > L$$



"G is efficient": $\text{poly}(\lambda)$

Q: where is λ ? $\hookrightarrow$ an ω - λ -indexed sequence

A: Technically, D is

$$D_\lambda = \{D_1, D_2, \dots\}$$
$$R_\lambda = \{R_1, \dots\}$$

Ex: $D_\lambda = \{0, 1\}^\lambda$
 $\uparrow$
length- λ bit strings

$$R_1 = \{0, 13^{2\lambda}\}$$
$$G: \{0,1\}^n \rightarrow \{0,1\}^{2n} \text{ - polytime TM}$$

In crypto, nearly everything is λ -indexed, so we'll omit the indices...

Security:

$G: S \rightarrow K$ is a PRG if $\forall$ probabilistic poly(λ) time (PPT) adversaries A ,

$$\text{PRG}_{\text{adv}}[A, G] = \left| \Pr_{A, s} [A(G(s)) = 1 : s \notin S] - \Pr_{A, r} [A(r) = 1 : r \notin R] \right| \leq \text{negl}(\lambda)$$

where

$f(\lambda) \leq \text{negl}(\lambda)$ if $\forall c \in \mathbb{N}$

$$f(\lambda) = O\left(\frac{1}{\lambda^c}\right)$$

ex: $2^{-\lambda} \leq \text{negl}(\lambda)$

why?

$$2^{-\lambda} = \lambda^{\log_{\lambda} 2^{-\lambda}}$$

$$= \lambda^{-\lambda \log_{\lambda} 2}$$

$$= \lambda^{-\log 2 \frac{\lambda}{\log \lambda}} \quad \left. \vphantom{\lambda^{-\log 2 \frac{\lambda}{\log \lambda}}} \right] \text{ grows past any constant } c$$

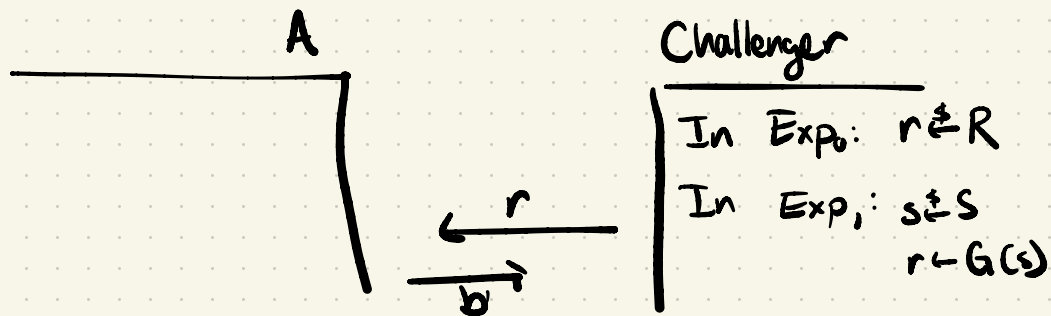
Facts:

$$f \leq \text{negl}, g \leq \text{poly} \Rightarrow f \cdot g \leq \text{negl}$$

$$f \not\leq \text{negl}, g \leq \text{poly} \Rightarrow f/g \not\leq \text{negl}$$

Back to PRGs.

PRG security can be seen as a game:



$$\text{PRG}_{\text{adv}}[A, G] = |\Pr[A \text{ outputs } 1 \text{ in } \text{Exp}_0] - \Pr[A \text{ outputs } 1 \text{ in } \text{Exp}_1]|$$

PRG practice, example 1:

PRGs in practice: ChaCha
or AES (counter mode)

Q: If $G(s \in \{0,1\}^L) \rightarrow r \in \{0,1\}^L$ is a PRG, is

$$G'(s \in \{0,1\}^L) \triangleq G(s) \oplus 1^L \stackrel{?}{=} \text{PRG?}$$

$\sim L$ repeats

A: Yes!

Let's prove it. (1st proof)

Goal: G is a PRG $\Rightarrow G'$ is a PRG

Contrapositive:

G' isn't a PRG $\Rightarrow G$ isn't a PRG

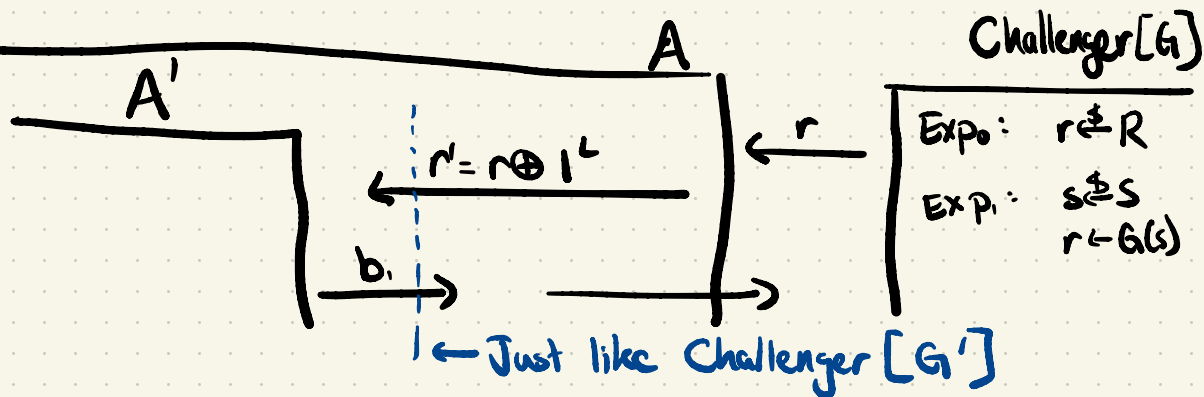
$\exists A', \text{PRGadv}[A', G'] \neq \text{negl} \Rightarrow \exists A, \text{PRGadv}[A, G] \neq \text{negl}$

fix A' s.t. $\text{PRGadv}[A', G'] \neq \text{negl}$.

goal: define an A s.t. $\text{PRGadv}[A, G] \neq \text{negl}$

$\rightarrow A$ will use A'

$\rightarrow A$ gets r , outputs $A'(r \oplus 1^L)$



Goal now: relate $A \sim A'$ in Exp_0 and Exp_1

Exp₁: A gets $r = G(s)$

A' gets $r' = G(s) \oplus 1^L = G'(s)$

$\Pr[A \rightarrow 1 \text{ in Exp}_1 \text{ (for } G)]$

$= \Pr[A' \rightarrow 1 \text{ in Exp}_1 \text{ (for } G')]$

Exp₀: A gets uniform $r \in \mathbb{R}$

A' gets $r = r \oplus 1^L$, also uniform

Fact: if a, b are independent RV's and a is uniform, $a \oplus b$ is uniform.

$\Pr[A \rightarrow 1 \text{ in Exp}_0] = \Pr[A' \rightarrow 1 \text{ in Exp}_0]$

So,

$\text{PRG}_{\text{adv}}[A, G]$

$= |\Pr[A \rightarrow 1 \text{ in Exp}_1] - \Pr[A \rightarrow 1 \text{ in Exp}_0]|$

$= |\Pr[A' \rightarrow 1 \text{ in Exp}_1] - \Pr[A' \rightarrow 1 \text{ in Exp}_0]|$

$= \text{PRG}_{\text{adv}}[A', G']$

$\neq \text{negl}$

result:

$\text{PRG}_{\text{adv}}[A, G] \neq \text{negl}$

$\Rightarrow G$ is not a PRG

✓

PRG practice 2:

Q: If $G(s) \rightarrow r$ is a PRG, is
 $G'(s) \triangleq (G(s), 1)$

A: No! (our 1st attack) $\leadsto$ all a 1 at the end

pf: no reduction needed

we will show $\exists$ a PPT A' such that:

$$\text{PRG}_{\text{adv}}[A', G'] \not\approx \text{negl}$$

Define $A'(r)$ to output r 's last bit

Lets compute

$$\begin{aligned} \Pr[A' \rightarrow 1 \text{ in } \text{Exp}_0] &= \Pr[A'(r) = 1 : r \xleftarrow{\$} R \times \{0, 1\}] && (\text{def. Exp}_0) \\ &= \Pr[(r \text{'s last bit}) = 1 : r \xleftarrow{\$} R \times \{0, 1\}] && (\text{def. } A') \\ &= 1/2 && (r \text{ uniform}) \end{aligned}$$

$$\begin{aligned} \Pr[A' \rightarrow 1 \text{ in } \text{Exp}_1] &= \Pr[A'(G(s) || 1) = 1 : s \xleftarrow{\$} S] && (\text{def. Exp}_1) \\ &= \Pr[1 = 1 : s \xleftarrow{\$} S] && (\text{def. } A') \\ &= 1 \end{aligned}$$

$$\text{So: } \text{PRG}_{\text{adv}}[A', G'] = |1/2 - 1| = 1/2 \not\approx \text{negl}$$

Recap:

- PRG definition: output indistinguishable from random
- $G(s)$ is a PRG $\Rightarrow G(s) \oplus 1^L$ is a PRG
 - pf: a reduction
- $G(s)$ is a PRG $\nRightarrow (G(s), 1)$ is a PRG.
 - pf: an attack

Next time:

- PRGs + hybrid arguments