CS 355 Topics in Cryptography

Lecture 1

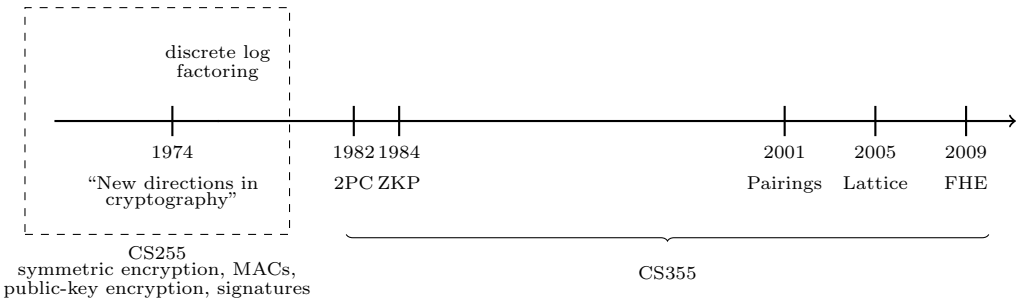
Edited by ZHENG Tianyu

23 September 2022

Outline

- What this course is about?
- Logistics, administrivia.
- Foundations of modern cryptography.

1 What this course is about?



We are at an "inflection" point in the development of cryptography:

1. Real, large-scale deployment of "fancy" cryptography (i.e beyond public-key cryptography)
   - Systems like Zcash (relies on pairing-based cryptography and zero-knowledge proof systems)
2. Potential threat of quantum computers requires re-thinking much of the existing public-key cryptography
   - Google recently ran pilot project implementing ring-LWE based key-exchange into Chrome (alongside traditional Diffie-Hellman)

- Ongoing NIST competition to standardize new post-quantum cryptography (expect 5-7 years to converge on new standards)

1.1 Course organization reflects these new developments:

1. Foundations of modern cryptography
2. Zero-knowledge and protocols
3. Post-quantum cryptography
4. Applications of cryptography

1.2 Our goals in this course:

1. Be your first course in advanced cryptography: teach you the foundations of modern cryptography and prepare you to get started with research in cryptography
2. Be your last course in advanced cryptography: give you a taste of the newest development in cryptography and enable you to build the next generation of systems using cutting-edge cryptography!

1.3 Logistics and Administrivia:

Skip.

1.4 Foundations of Modern Cryptography:

Modern cryptography is the study of hardness:

There is some task that should be "easy" for an honest party, but "difficult" for an adversary,

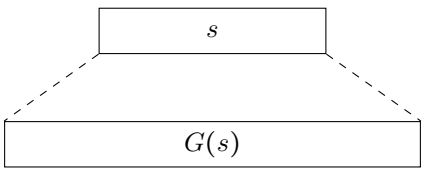
Example 1. *Encryption scheme: given knowledge of the key, it is easy to decrypt, but without the secret key, is is difficult to decrypt.*

Question:How do we model this mathematically?

This will be very important to develop and understand more advanced concepts. In this lecture, we will focus on symmetric primitives.

For our first example, consider a pseudorandom generator (PRG).

Recall: A PRG takes a short seed s and expands it into a long "random-looking" string:



$s \in \{0,1\}^\lambda$, λ is the length of the seed (i.e., security parameter)

$G(s) \in \{0,1\}^{\ell(\lambda)}$, often called the "stretch" of the PRG

Question: Why should $\ell(\lambda) > \lambda$

Question: What does it mean to be "random-looking"?

Intuitively: No efficient algorithm should be able to distinguish it from a truly-random string.

Formally: "efficient" (for a PRG, seed length = security parameter) will mean polynomial time (in the security parameter).

Define distinguishing algorithm as one that takes a string (either output of PRG or truly random string) and guesses whether the string is output of PRG or actual random string.

Definition 1. *A PRG* $G : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$ *is secure if for all (possibly randomized) algorithms* $\mathcal{A}$ *running in time* $\text{poly}(\lambda)$*:*

$$\left|\Pr[s \xleftarrow{R} \{0,1\}^\lambda : \mathcal{A}(G(s)) = 1] - \Pr[t \xleftarrow{R} \{0,1\}^{\ell(\lambda)} : \mathcal{A}(t) = 1]\right| < \text{negl}(\lambda),$$

where s is sampled uniformly from $\{0,1\}^\lambda$, and

1. $\Pr[s \xleftarrow{R} \{0,1\}^\lambda : \mathcal{A}(G(s)) = 1]$: adversary sees output of PRG on random seed.
2. $\Pr[t \xleftarrow{R} \{0,1\}^{\ell(\lambda)} : \mathcal{A}(t) = 1]$: adversary sees totally random string.
3. $\text{negl}(\lambda)$: a function $f(\lambda)$ is negligible in λ if $f(\lambda) = \mathcal{O}(\frac{1}{\lambda^c})$ for all constants $c \in N$.

Intuitively: behavior of algorithm $\mathcal{A}$ does not vary much on PRG outputs and truly random outputs.

Oftentimes, we define following variables:

$$W_0 = \Pr[s \xleftarrow{R} \{0,1\}^\lambda : \mathcal{A}(G(s)) = 1], \quad \text{"pseudorandom"}$$
$$W_1 = \Pr[t \xleftarrow{R} \{0,1\}^{\ell(\lambda)} : \mathcal{A}(t) = 1]. \quad \text{"random"}$$

The PRG distinguishing advantage is then $\text{PRGAdv}[\mathcal{A}, G] = |W_0 - W_1|$.

We can also view this definition through the lens of computational indistinguishability.

Definition 2. *Let* $\lambda \in \mathbb{N}$ *be a security parameter. Let* $D_1 = \{D_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ *and* $D_2 = \{D_{2,\lambda}\}_{\lambda\in\mathbb{N}}$ *be collections (ensembles) of distributions indexed by* λ*. Then,* D_1 *and* D_2 *are computationally indistinguishable (denoted* $D_1 \approx^c D_2$*) if for all efficient adversaries* $\mathcal{A}$*,*

$$|\Pr[x_1 \leftarrow D_{1,\lambda} : \mathcal{A}(1^\lambda, x_1) = 1] - \Pr[x_2 \leftarrow D_{2,\lambda} : \mathcal{A}(1^\lambda, x_2) = 1]| = \text{negl}(\lambda).$$

Definition 3. *(PRG security definition)* $G : \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)}$ *is a secure PRG if* $D_1 \approx^c D_2$ *where*

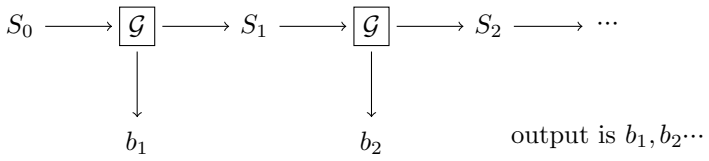
$$D_1 = \{s \xleftarrow{R} \{0,1\}^\lambda : G(s)\},$$
$$D_2 = \{t \xleftarrow{R} \{0,1\}^{\ell(\lambda)} : t\}.$$

Question: Do PRGs exist?

Unknown! Requires (minimally) resolving $\mathcal{P}$ vs. $\mathcal{NP}$. But if one-way functions (OWFs) exist, then we can build PRG that expand by a single bit: $\ell(\lambda) = \lambda + 1$ (Will explore this on HW1, Problem 2)

Question: How to go from one-bit PRG to multi-bit PRG?

Blum-Micali PRG: Suppose we have PRG $\mathcal{G}$ that expands by 1-bit:



We can iterate m times to obtain m output bits.

Question: Is this construction secure?

To prove security, we will use a "hybrid" argument.

Suppose we iterate Blum-Micali m times. We need to show that the following distributions are computationally indistinguishable:

$$\begin{aligned} D_1 &= \{s_0 \xleftarrow{R} \{0,1\}^\lambda : H(s_0)\}, \\ D_2 &= \{t \xleftarrow{R} \{0,1\}^m : t\}, \end{aligned}$$

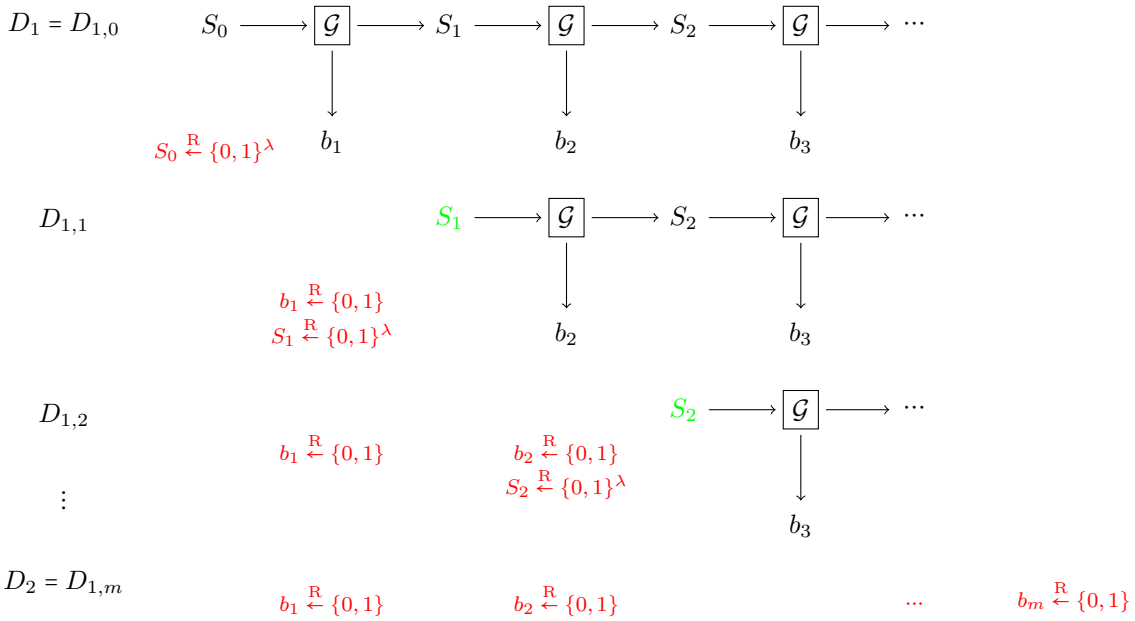
where $H(s_0)$ is m iterations of Blum-Micali.

Define intermediate distributions $D_{1,i}$ for $i = 0, ..., m$ as following figure.

Let $p_{1,0} = \Pr[x \leftarrow D_{1,0} : \mathcal{A}(x) = 1]$ ... $p_{1,m} = \Pr[x \leftarrow D_{1,m} : \mathcal{A}(x) = 1]$ then,

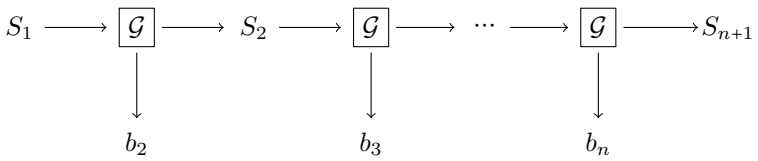
$$\begin{aligned} \text{PRGAdv}[\mathcal{A}, H] &= |p_{1,0} - p_{1,m}| \\ &= |p_{1,0} - p_{1,1} + p_{1,1} - p_{1,2} + \cdots + p_{1,m} - p_{1,m}| \\ &\leq |p_{1,0} - p_{1,1}| + |p_{1,1} - p_{1,2}| + \cdots + |p_{1,m} - p_{1,m}| \text{ by triangle inequality} \end{aligned}$$

Observe: Quantity $|p_{1,0} - p_{1,1}|$ is essentially distinguishing advantage of some algorithm for $\mathcal{G}$.



More precisely, suppose $|p_{1,0} - p_{1,1}| = \epsilon$. We use $\mathcal{A}$ to build distinguisher $\mathcal{B}$ for $\mathcal{G}$ such that distinguisher $\mathcal{B}$ works as follows:

1. On input a string $z \in \{0,1\}^{\lambda+1}$, $\mathcal{B}$ parses $z = (b_1, s_1)$
2. Algorithm $\mathcal{B}$ computes



   and invokes $\mathcal{A}$ on inputs $b_1, ..., b_n$

3. Algorithm $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs

Tow case to consider:

- if $(b_1, s_1) \leftarrow G(s)$ where $s \leftarrow \{0,1\}^\lambda$, then $\mathcal{A}$ outputs 1 with probability $p_{1,0}$ (by definition of $D_{1,0}$)
- if $(b_1, s_1) \xleftarrow{R} \{0,1\}^{\lambda+1}$, $\mathcal{A}$ outputs 1 with probability $p_{1,1}$ (by definition of $D_{1,1}$)

Thus, $\mathrm{PRGAdv}[\mathcal{B}, \mathcal{G}] = |p_{1,0} - p_{1,1}| = \epsilon$. ■

Conclusion: We can bound each $|p_{1,i} - p_{1,i+1}|$ by $\mathrm{PRGAdv}[\mathcal{B}, \mathcal{G}]$ for some efficient distinguisher $\mathcal{B}$. Thus,

$$\mathrm{PRGAdv}[\mathcal{A}, \mathcal{H}] \leq m \cdot \mathrm{PRGAdv}[\mathcal{B}, \mathcal{G}],$$

since $\mathcal{G}$ is a secure PRG and $\mathcal{B}$ is efficient,

$$\mathrm{PRGAdv}[\mathcal{B}, \mathcal{G}] = \mathrm{negl}(\lambda),$$

so as long as $m = \mathrm{poly}(\lambda)$,

$$\mathrm{PRGAdv}[\mathcal{A}, \mathcal{H}] \leq m \cdot \mathrm{PRGAdv}[\mathcal{B}, \mathcal{G}] = m \cdot \mathrm{negl}(\lambda) = \mathrm{negl}(\lambda). \qquad \blacksquare$$

General hybrid argument approach: to argue indistinguishability of two distributions:

1. Identify sequence of intermediate distributions between the target distributions
2. Argue that each consecutive pair of hybrid distributions are computationally indistinguishable

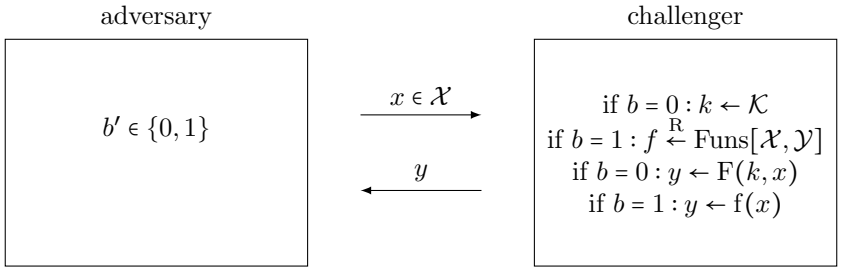
More abstractly: if $D_1 \approx^c D_2 \approx^c \cdots \approx^c D_m$ and $m = \mathrm{poly}(\lambda)$, then $D_1 \approx^c D_m$ (computational indistinguishability is essentially transitive).

Other symmetric primitives (review):

Pseudorandom functions (PRFs): A PRF $\mathrm{F} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ (strictly speaking, $\mathcal{K}, \mathcal{X}, \mathcal{Y}$ are all ensembles indexed by λ) on a key space $\mathcal{K}$, domain $\mathcal{X}$ and range $\mathcal{Y}$ if for all efficient adversaries $\mathcal{A}$,

$$\mathrm{PRGAdv}[\mathcal{A}, \mathrm{F}] = |\Pr[W_0 = 1] - \Pr[W_1 = 1]| = \mathrm{negl}(\lambda)$$

where for $b \in \{0, 1\}$, we define W_b as output of $\mathcal{A}$ in the following experiment:



In particular, $\mathcal{A}$ can not distinguish outputs of PRF from that of a truly random function.

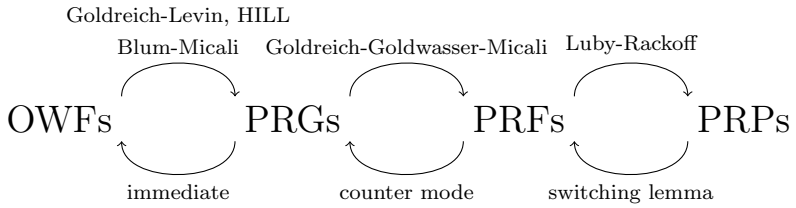
Pseudorandom permutations (PRPs): Similar to PRFs except replace function with permutation.

One-way functions (OWFs): A function $f : \mathcal{X}_\lambda \leftarrow \mathcal{Y}_\lambda$ is one-way if for all efficient adversaries $\mathcal{A}$:

$$\Pr[x \leftarrow \mathcal{X}_\lambda : f(\mathcal{A}(f(x))) = x] = \text{negl}(\lambda)$$

Intuitively: One-way functions are difficult to invert.

Connections between symmetric primitives:

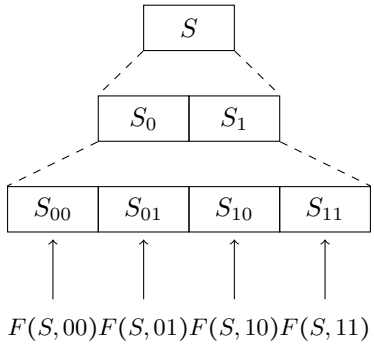


Implication: OWFs is minimal assumption for symmetric cryptography.

[If there is time: Goldreich-Goldwasser-Micali: PRF from PRG]
Suppose we have a length doubling PRG $\mathcal{G} : \{0,1\}^\lambda \leftarrow \{0,1\}^{2\lambda}$

Construction:	$\mathcal{K}_\lambda = \{0,1\}^\lambda$
	$\mathcal{X}_\lambda = \{0,1\}^{n(\lambda)}$
	$\mathcal{Y}_\lambda = \{0,1\}^\lambda$
Write:	$\mathcal{G}(s) \leftarrow (s_0, s_1) = (\mathcal{G}_0(s), \mathcal{G}_1(s))$
Then,	PRF value at $x_1, ..., x_n$ is $\mathcal{G}_{x_n}(\mathcal{G}_{x_{n-1}}(\cdots \mathcal{G}_{x_1}(s) \cdots))$

Picture (when $n = 2$):



Security proof is another hybrid argument (left as exercise)