Game sheet

Gidon Rosalki

February 3, 2026

Notice: If you find any mistakes, please open an issue at `https://github.com/robomarvin1501/notes_intro_to_crypto`

1 Perfect secrecy

Definition 1.1 (Perfect secrecy)**.** *A symmetric key encryption scheme* $\Pi = (KeyGen, Enc, Dec)$ *is* ***perfectly secret*** *if for every distribution over* $\mathcal{M}$*, and for every* $m \in \mathcal{M}$*, and for every* $c \in \mathcal{C}$ *it holds that*

$$Pr\left[M = m | C = c\right] = Pr\left[M = m\right]$$

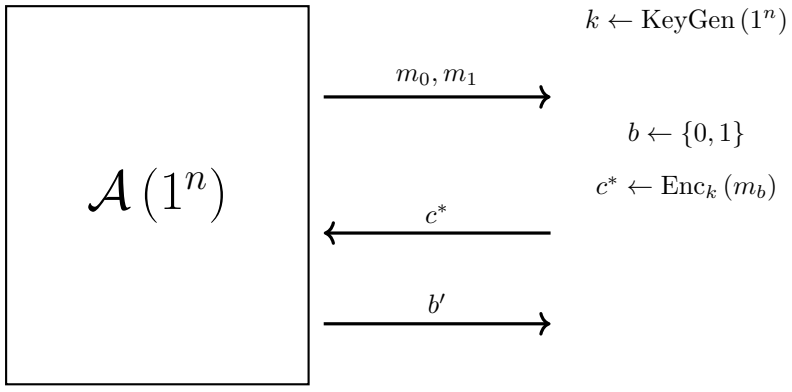
That is, the probability that some plaintext is the plaintext given the ciphertext, is the same as the probability that some plaintext is the plaintext, with no priors whatsoever.

2 Indistinguishable encryption

Definition 2.1 (Indistinguishable encryption)**.** Π *has* ***indistinguishable encryptions*** *if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $v\left(\cdot\right)$ *such that*

$$\mathbb{P}\left[IND_{\Pi,\mathcal{A}}\left(n\right) = 1\right] \leq \frac{1}{2} + v\left(n\right)$$

where the probability is taken over the random coins used by $\mathcal{A}$*, and by the experiment.*



$$\text{IND}_{\Pi,\mathcal{A}}\left(n\right) = \begin{cases} 1, & \text{if } b' = b \\ 0, & \text{otherwise} \end{cases}$$

3 PRGs

Definition 3.1 (PRG)**.** *Let* $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ *be a polynomial-time computable function, and let* $l\left(\cdot\right)$ *be a polynomial such that for any input* $s \in \{0,1\}^n$*, we have* $G\left(s\right) \in \{0,1\}^{l(n)}$*. Then,* G *is a* ***pseudorandom generator*** *if the following two conditions hold:*

- *Expansion:* $l\left(n\right) > n$
- *Pseudorandomness: For every PPT "distinguisher"* $\mathcal{D}$*, there exists a negligible function* $v\left(\cdot\right)$ *such that*

$$\left| \Pr_{s \leftarrow \{0,1\}^n}\left[\mathcal{D}\left(G\left(s\right)\right) = 1\right] - \Pr_{r \leftarrow \{0,1\}^{l(n)}}\left[\mathcal{D}\left(r\right) = 1\right] \right| \leq v\left(n\right)$$

So, the probability that the distinguisher may tell the difference between the output of the PRG, and truly random noise, is less than the output of the negligible function for that length of input.

4 Semantic security

Definition 4.1 (Semantically secure)**.** Π *is* ***semantically secure*** *if for every adversary* $\mathcal{A}$ *there exists a PPT "simulator"* $\mathcal{S}$ *such that for every efficiently sampleable plaintext distribution* $M = \{M_n\}_{n\in\mathbb{N}}$*, and all polynomial-time computable functions* f *and* h*, there exists a negligible function* $v(\cdot)$ *such that*

$$|\Pr[\mathcal{A}(1^n, Enc_k(m), h(m)) = f(m)] - \Pr[\mathcal{S}(1^n, h(m)) = f(m)]| \leq v(n)$$

where $k \leftarrow KeyGen(1^n)$ *and* $m \leftarrow M_n$

Or in other words, whatever you can learn from the encryption, can also be efficiently learnt *without* the encryption, or most simply, the ciphertext teaches us **nothing**. Π is *semantically secure* **if and only if** it has *indistinguishable encryption.*

5 One way functions

Definition 5.1. *A polynomial-time computable function* $f : \{0,1\}^* \to \{0,1\}^*$ *is* ***one way*** *if for any PPT* $\mathcal{A}$*, and negligible function* $v(\cdot)$

$$\Pr_{y\leftarrow f(U_n)}\left[\mathcal{A}(1^n, y) \in f^{-1}(y)\right] \leq v(y)$$

Easy to compute, hard to invert.

6 Computational Indistinguishability

Definition 6.1 (Computationally indistinguishable)**.** *Two probability distributions* $X = \{X_n\}_{n\in\mathbb{N}}$ *and* $Y = \{Y_n\}_{n\in\mathbb{N}}$ *are* ***computationally indistinguishable*** *if for every PPT distinguisher* D *there exists a negligible function* $v(\cdot)$ *such that*

$$\left|\Pr_{x\leftarrow X_n}[D(1^n, x) = 1] - \Pr_{y\leftarrow Y_n}[D(1^n, y) = 1]\right| \leq v(n)$$

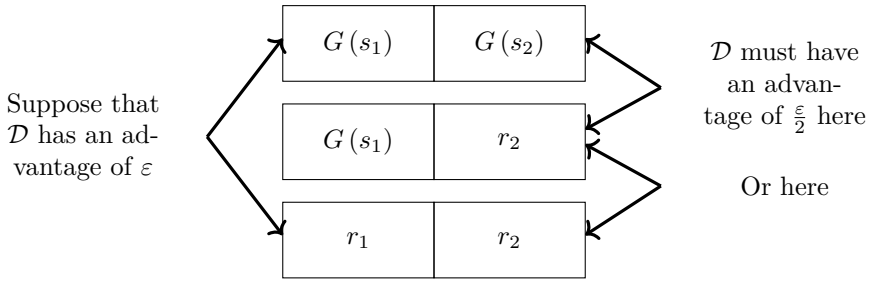
This is denoted $X \approx^c Y$

7 Hybrid argument

This is a complicated technique, so we shall present an example.

Theorem 1. *Let* $G : \{0,1\}^n \to \{0,1\}^{4n}$ *be a PRG, then* $H(s_1, s_2) = G(s_1)||G(s_2)$ *is a PRG.*

Proof . Our paradigm for this kind of proof is *reduction* via a *hybrid argument.*
Reduction: Given a distinguisher D, for H, construct a distinguisher A for G.
Hybrid argument: Let us suppose that between $G(s_1), G(s_2)$ D has advantage ε. Let us create a new PRG, that given s_1, s_2, ignores s_2, and returns $G(s_1), r_2$. So, between $G(s_1), G(s_2)$ and $G(s_1), r_2$, it holds that D has at least the advantage $\frac{\varepsilon}{2}$, or between $G(s_1), r_2$ and r_1, r_2 it holds that D has the advantage of at least $\frac{\varepsilon}{2}$.



So:

$$\begin{aligned}\varepsilon &\leq |\mathbb{P}[D(G(s_1)||G(s_2)) = 1] - \mathbb{P}[D(r_1||r_2) = 1]| \\ &\leq |\mathbb{P}[D(G(s_1)||G(s_2)) = 1] - \mathbb{P}[D(G(s_1)||r_2) = 1]| + |\mathbb{P}[D(G(s_1)||r_2) = 1] - \mathbb{P}[D(r_1||r_2) = 1]|\end{aligned}$$

Let us define A, which on input $z \in \{0,1\}^{4n}$ with sample $s_1 \leftarrow \{0,1\}^n$ and output $D(G(s_1)\|z)$. In this case, we have created an adversary that distinguishes between the first 2 cases based off the difference of $G(s_2)$ and r_2. We may similarly create a second adversary that performs the same, and outputs $D(Z\|r_2)$. Since *one* of these transitions must be distinguishable with an advantage of at least $\frac{\varepsilon}{2}$, we have found an adversary A for G, which is a contradiction to the given that G is a PRG. $\square$

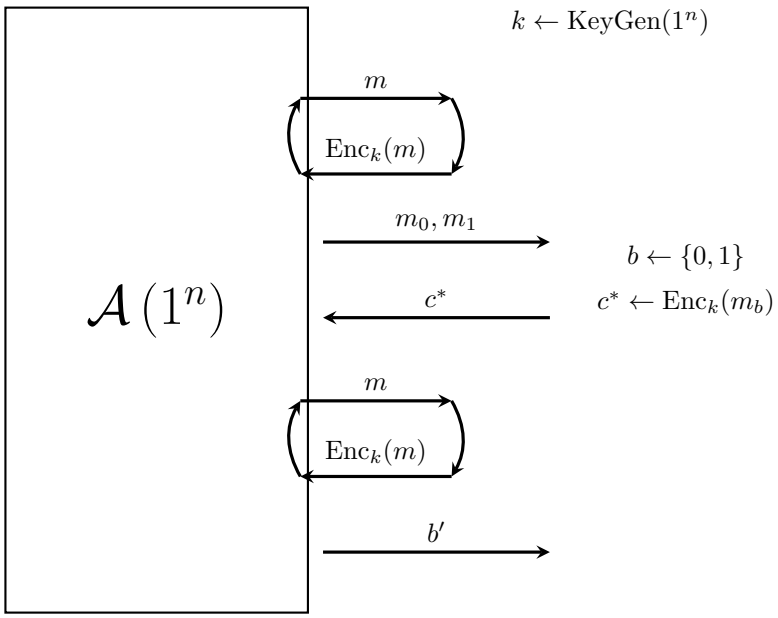
8 Chosen Plaintext Attack (CPA)

We can modify Indistinguishable Encryption such that $\mathcal{A}$ may request any number of encryptions (From an oracle), before it hands over the two messages between which it must distinguish:

Definition 8.1 (IND-CPA)**.** Π *has indistinguishable encryptions under a chosen-plaintext attack if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $v(\cdot)$ *such that*

$$\Pr\left[IND^{CPA}_{\Pi,\mathcal{A}}(n) = 1\right] \leq \frac{1}{2} + v(n)$$

This is to say, that the probability of winning the CPA game (described below) is 50%, plus negligible.



$$\text{IND}^{\text{CPA}}_{\Pi,\mathcal{A}}(n) = \begin{cases} 1, & \text{if } b' = b \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

9 Pseudorandom Functions

Definition 9.1 (PRF)**.** *An efficiently computable keyed function*

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{l(n)}$$

is ***pseudorandom*** *if for every PPT distinguisher $\mathcal{D}$ there exists a negligible function $v(\cdot)$ such that*

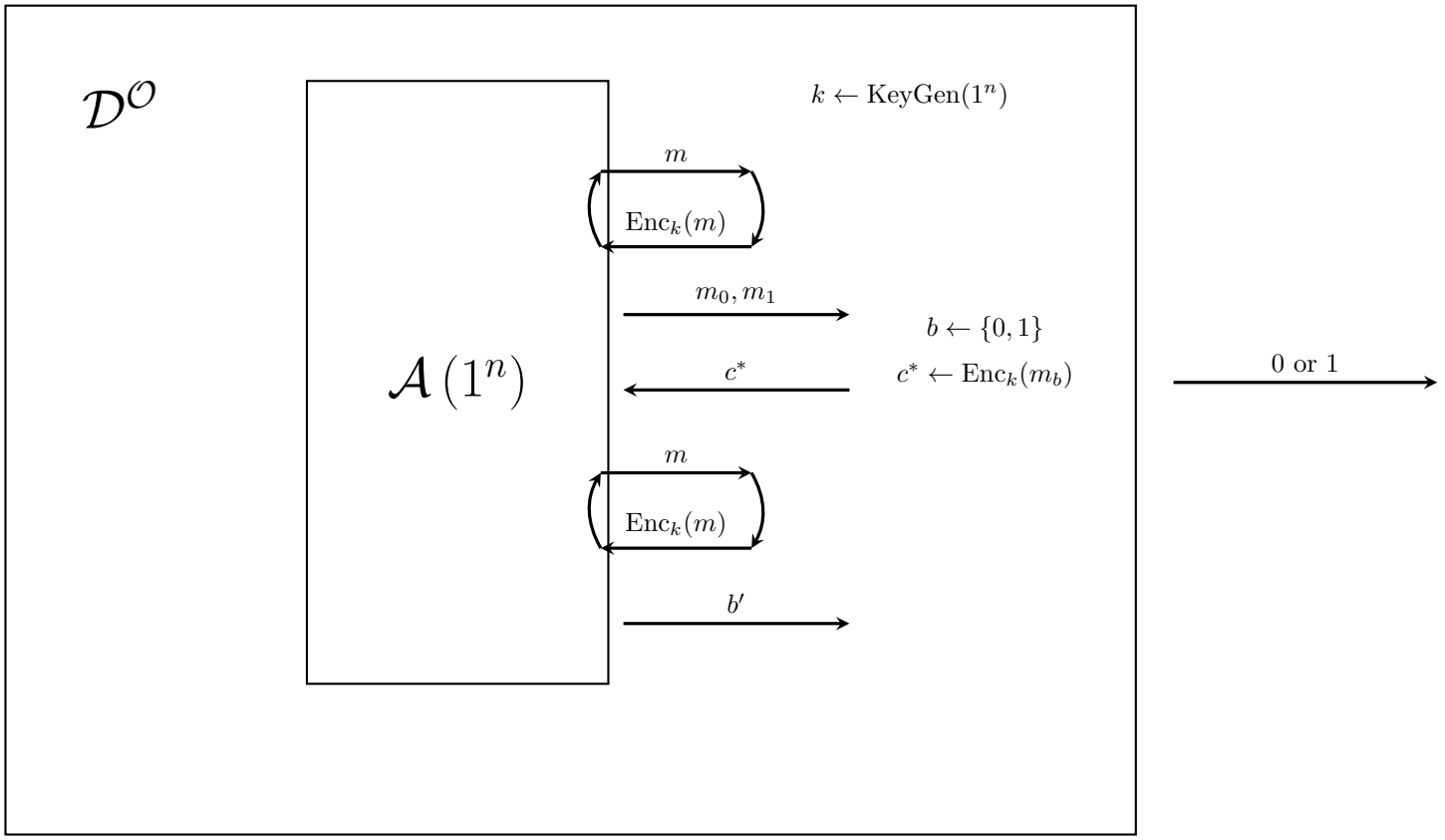
$$\left|\Pr\left[\mathcal{D}^{F_k(\cdot)}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{h(\cdot)}(1^n) = 1\right]\right| \leq v(n)$$

where $k \leftarrow \{0,1\}^n$ and $h \leftarrow Func_{n \to l}$

The methodology for using PRFs is as follows:

1. Prove security assuming a truly random function is used
2. Prove that if an adversary can break the scheme when PRF is used, then it can be used to distinguish the PRF from a truly random function

We may consider Enc to be, for example something that returns $(r, \mathcal{O}(r) \oplus m_b)$, and thus try and show if this is a CPA secure scheme or not. For example, for the theorem *If F is a PRF, then Π_F is CPA-Secure.* For the truly random function h, Π_h is secure, so we may show that Π_h is indistinguishable from Π_F, by contradiction that finds that Π_F is not a PRF.



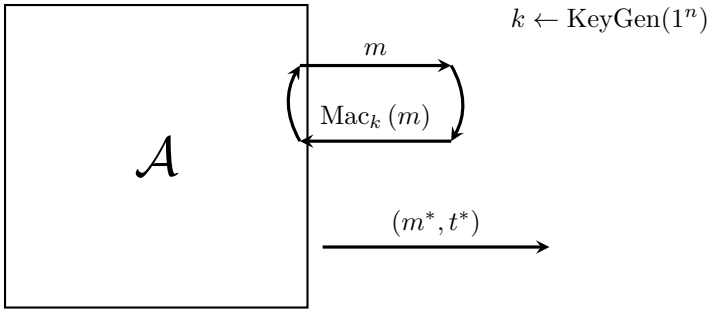
$$\text{IND}^{\text{CPA}}_{\Pi,\mathcal{A}}(n) = \begin{cases} 1, & \text{if } b' = b \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

10 MACs

Definition 10.1 (MAC scheme)**.** *A MAC (Message Authentication Code) scheme* $\Pi = (Gen, Mac, Vrfy)$ *is secure if for every PPT adversary* $\mathcal{A}$*, there exists a negligible function* $v(\cdot)$ *such that*

$$\Pr\left[MacForge_{\Pi,\mathcal{A}}(n) = 1\right] \leq v(n)$$

This is to say, it is very hard for a PPT adversary to create a new message, with a correct MAC.



$$\text{Let } \mathcal{Q} = \text{ the set of all queries asked by } \mathcal{A} \tag{3}$$

$$\text{MacForge}_{\Pi,\mathcal{A}}(n) = \begin{cases} 1, & \text{if } \text{Vrfy}_k(m^*, t^*) = 1 \wedge m^* \notin \mathcal{Q} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

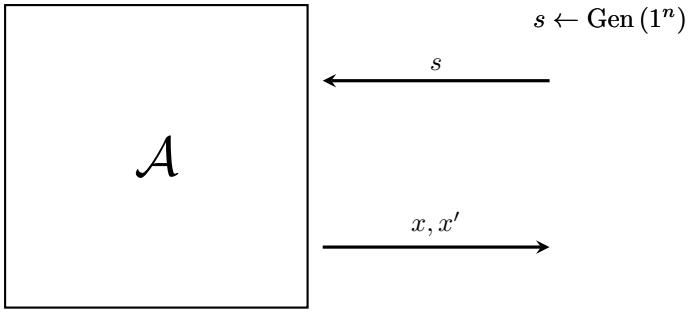
Note that this does **not** prevent replay attacks!

11 CRHF

Definition 11.1 (Collision Resistant)**.** Φ *is* ***collision resistant*** *if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $v(\cdot)$ *such that*

$$\Pr\left[HashColl_{\Phi,\mathcal{A}}(n) = 1\right] \leq v(n)$$

We may describe HashColl as follows:



$$\text{HashColl}_{\Phi,\mathcal{A}}(n) = \begin{cases} 1, & \text{if } H_s(x) = H_s(x') \wedge x \neq x' \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

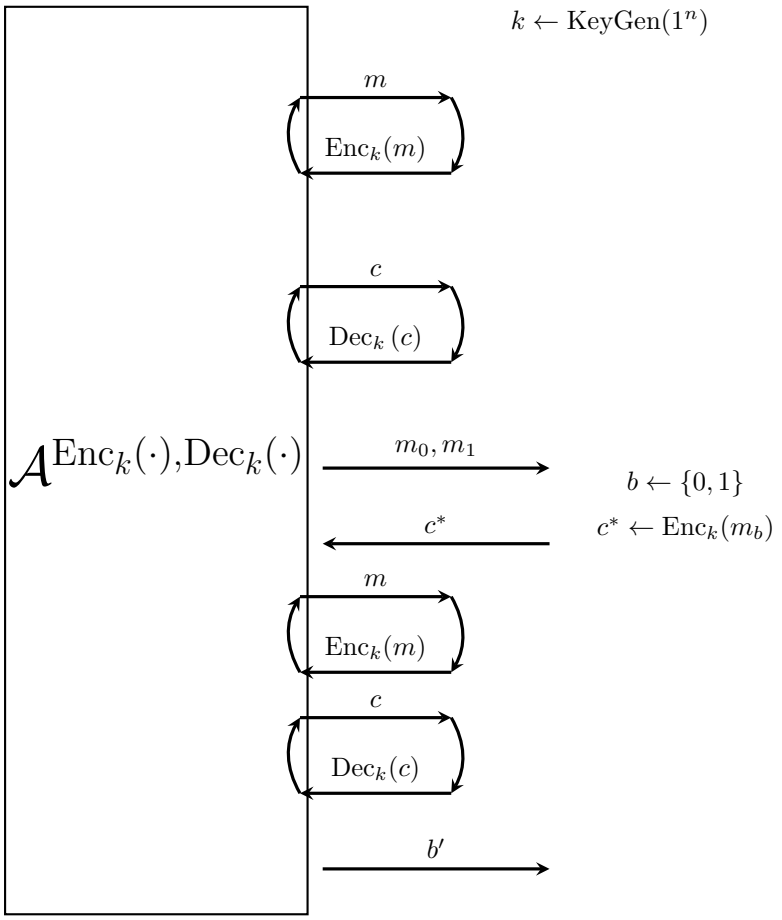
12 CCA

Definition 12.1 (CCA-IND)**.** Π *has indistinguishable encryptions under a chosen-ciphertext attack if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $v(\cdot)$ *such that*

$$\Pr\left[IND^{CCA}_{\Pi,\mathcal{A}}(n)=1\right]\leq\frac{1}{2}+v(n)$$

In this case, we may also say that Π *is CCA-secure.*

Note that CCA implies authenticity, since given $\mathrm{Enc}_k(m)$, it is hard to generate $\mathrm{Enc}_k(m')$ for a "related" m' (such as $m'=m+1$).



$$\mathcal{Q}=\text{ set of all decryption queries asked by }\mathcal{A} \tag{6}$$

$$\mathrm{IND}^{\mathrm{CCA}}_{\Pi,\mathcal{A}}(n)=\begin{cases}1, & \text{if } b'=b\wedge c^*\notin\mathcal{Q}\\ 0, & \text{otherwise}\end{cases} \tag{7}$$

13 Key Agreement

Definition 13.1 (Correctness)**.** Π *is a* ***key agreement protocol*** *if there exists a negligible function* $v(n)$ *such that for all* $n\in\mathbb{N}$

$$\Pr_{r_A,r_B}\left[K_A(1^n,r_A,r_B)\neq K_B(1^n,r_A,r_B)\right]\leq v(n)$$

This is to say, that K_i generates a different key given the same inputs with an exceedingly low probability.
The important thing to note here is that Eve is eavesdropping the communication channel, and should not learn **any** information on the resulting key. Specifically, from Eve's point of view, the key should be "as good as" an independently chosen key.

Definition 13.2 (Security)**.** *A key agreement protocol* Π *is* ***secure*** *if*

$$\left(Transcript_\Pi(1^n,r_A,r_B),K_A(1^n,r_A,r_B)\right)\approx^c\left(Transcript_\Pi(1^n,r_A,r_B),K\right)$$

Where $r_A,r_B\leftarrow\{0,1\}^*$, $K\leftarrow\mathcal{K}_n$ *are sampled independently and uniformly.*

In order to create such a protocol, it is important to first remember the definition of *computational indistinguishability.* Two probability distributions are computationally indistinguishable if no efficient algorithm can tell them apart:

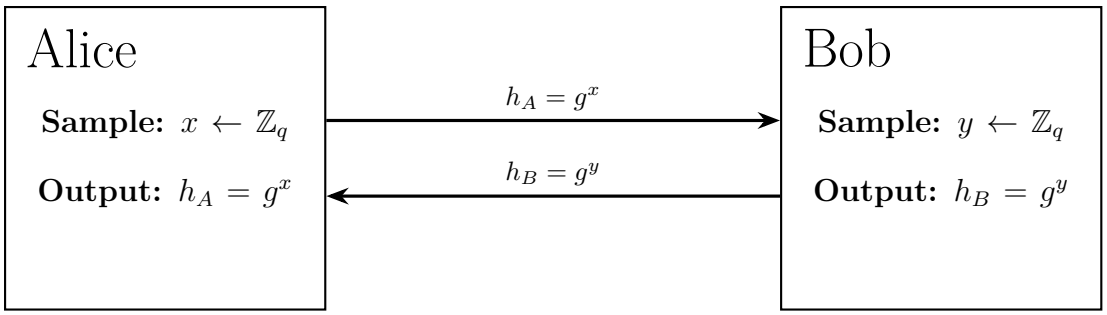
Definition 13.3 (Computationally indistinguishable)**.** *Two probability ensembles* $X = \{X_n\}_{n\in\mathbb{N}}, Y = \{Y_n\}_{n\in\mathbb{N}}$ *are* ***computationally indistinguishable*** *if for all PPT distinguishers* $\mathcal{D}$ *there exists a negligible function* $v(\cdot)$ *such that*

$$|\Pr[\mathcal{D}(1^n, x) = 1] - \Pr[\mathcal{D}(1^n, y) - 1]| \leq v(n)$$

Where $x \leftarrow X_n,\ y \leftarrow Y_n$

13.1 Diffie-Hellman

Let $\mathcal{G}$ be a PPT algorithm that on input 1^n, outputs $(\mathbb{G}, q, g)$, where $\mathbb{G}$ is a cyclic group of order q, that is generated by g, and q is an $n$bit prime. Let us assume that $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ is generated, and known to both parties (a publicly published one in the world).



Shared key: $K_A = (h_B)^x = g^{xy}$ Shared key: $K_B = (h_A)^y = g^{xy}$

$$K_A = (h_B)^x = (g^y)^x = (g^x)^y = (h_A)^y = K_B$$

So, Alice samples $x \leftarrow \mathbb{Z}_q$, and then computes $h_A = g^x$, which she sends to Bob. Similarly, Bob samples $y \leftarrow \mathbb{Z}_q$, computes $h_B = g^y$, which he sends to Alice. Alice then outputs $K_A = (h_B)^x$, and Bob outputs $K_B = (h_A)^y$.

Definition 13.4 (The Decisional Diffie Hellman (DDH) Assumption)**.** *For every PPT algoirithm* $\mathcal{A}$ *there exists a negligible function* $v(\cdot)$ *such that*

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xz}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1]| \leq v(n)$$

Where $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$*, and* $x, y, z \leftarrow \mathbb{Z}_q$

Effectively, they made an assumption that it is secure, and it has still not been broken. If you break it, you will get the Turing prize. Sadly, unlike Computability and Complexity, no guarantees of 100% in the course.

Definition 13.5 (Computational Diffie-Hellman Assumption)**.** *For every PPT algorithm* $\mathcal{A}$*, there exists a negligible function* $v(\cdot)$ *such that*

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y) = g^{xy}]| \leq v(n)$$

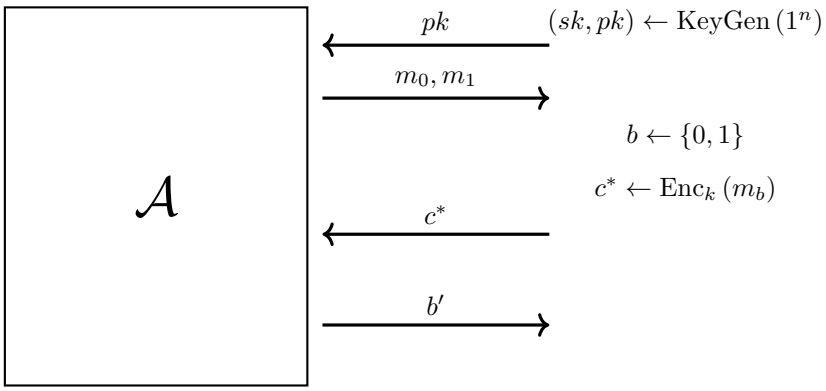
Where $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$*, and* $x, y \leftarrow \mathbb{Z}_q$

If you can solve CDH, then you can also solve DDH, so therefore DDH is a more secure assumption.

14 Public Key Encryption

Definition 14.1 (IND-CPA)**.** Π *has indistinguishable encryptions under a chosen-plaintext attack if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $v(\cdot)$ *such that*

$$\Pr\left[IND^{CPA}_{\Pi,\mathcal{A}}(n) = 1\right] \leq \frac{1}{2} + v(n)$$



$$\text{IND}^{\text{CPA}}_{\Pi,\mathcal{A}}(n) = \begin{cases} 1, & \text{if } b' = b \\ 0, & \text{otherwise} \end{cases}$$

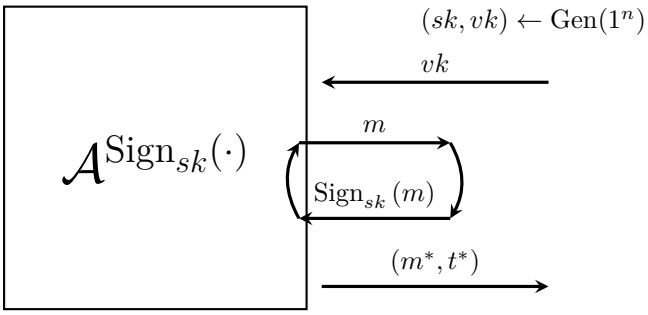
We will note that this is CPA, despite not having the oracle access, because $\mathcal{A}$ may function as its own oracle, since access to the public key means that $\mathcal{A}$ may encrypt any message that it wants.

15 Digital Signatures

Definition 15.1. Π *is* ***existentially unforgeable against an adaptive chosen message attack*** *if for every PPT adversary* $\mathcal{A}$*, there exists a negligible function* $v(\cdot)$ *such that*

$$\Pr\left[SigForge_{\Pi,\mathcal{A}}(n) = 1\right] \leq v(n)$$

Where the SigForge game is:



$$\text{Let } \mathcal{Q} = \text{ the set of all queries asked by } \mathcal{A} \tag{8}$$

$$\text{SigForge}_{\Pi,\mathcal{A}}(n) = \begin{cases} 1, & \text{if } \text{Vrfy}_{vk}(m^*, \sigma^*) = 1 \wedge m^* \notin \mathcal{Q} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

16 Interactive Proofs

Definition 16.1 (Interactive proof system)**.** *An* ***interactive proof system*** *for a language* L *is a protocol* $\langle \mathcal{P}, \mathcal{V} \rangle$ *where* $\mathcal{V}$ *is computable in probabilistic polynomial time, and the following holds:*

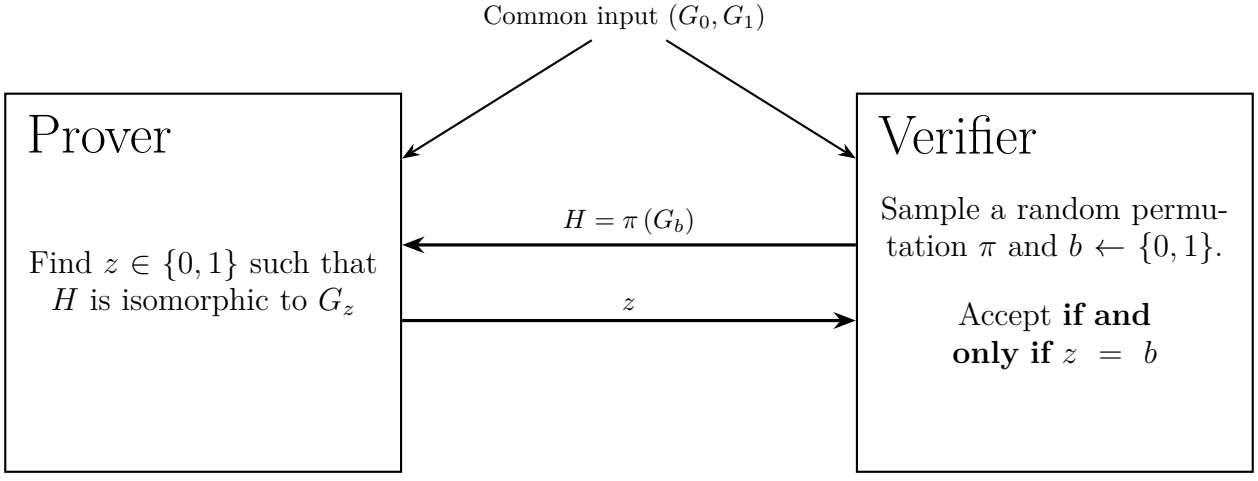
- *Completeness: For every* $x \in L$*:*
$$\Pr_{r_{\mathcal{P}}, r_{\mathcal{V}}}\left[out_{\mathcal{V}}\left[\langle \mathcal{P}, \mathcal{V} \rangle(x)\right] = Accept\right] = 1$$
- *Soundness: For every* $x \notin L$*, and for every computationally unbounded* $\mathcal{P}^*$*:*
$$\Pr_{r_{\mathcal{P}}, r_{\mathcal{V}}}\left[out_{\mathcal{V}}\left[\langle \mathcal{P}^*, \mathcal{V} \rangle(x)\right] = Accept\right] \leq \frac{1}{2}$$

We will state that **IP** is the class of all languages with an interactive proof system. IP contains NP, and in fact, $IP = PSPACE$. We can reduce the soundness error from $\frac{1}{2}$ to ε with $\log\left(\frac{1}{\varepsilon}\right)$ independent repetitions.

Behold, an example of an interactive proof:

Definition 16.2 (Isomorphic)**.** *Two graphs* $G_0 = (V_0, E_0)$*, and* $G_1 = (V_1, E_1)$ *are* ***isomorphic*** *if there exists a one to one mapping* $\pi : V_0 \to V_1$ *such that* $(u, v) \in E_0 \Leftrightarrow (\pi(u), \pi(v)) \in E_1$ *for every* $e, v \in V_0$

We can define the set of isomorphic graphs $GI = \{(G_0, G_1) : G_0 \text{ is isomorphic to } G_1\} \in NP$. Similarly, we can define the other class of graphs that are **not** isomorphic: $GNI = \{(G_0, G_1) : G_0 \text{ is } \textbf{not} \text{ isomorphic to } G_1\} \in NP$. This class is not known to be in NP.

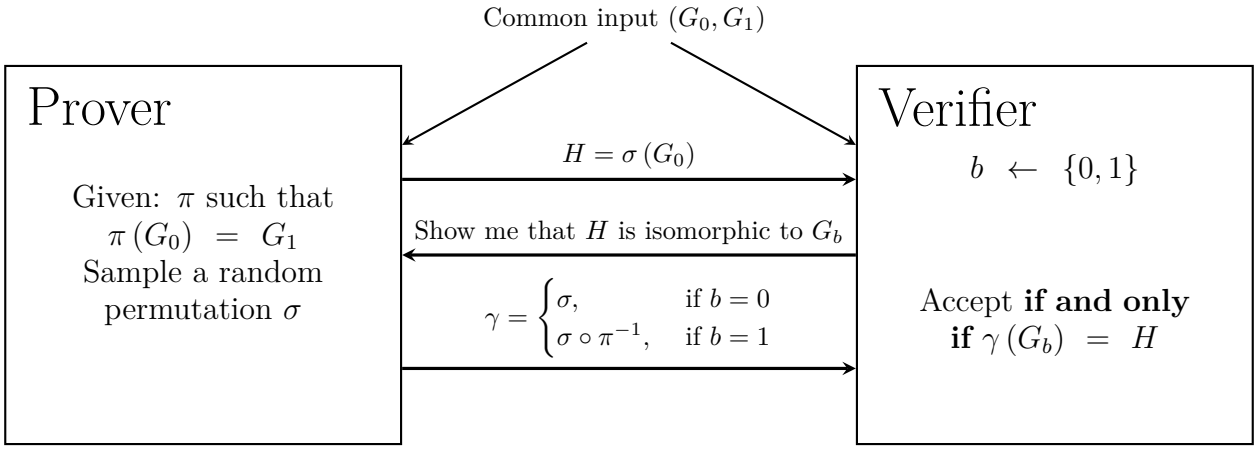


Proof method

17 Zero Knowledge Proofs

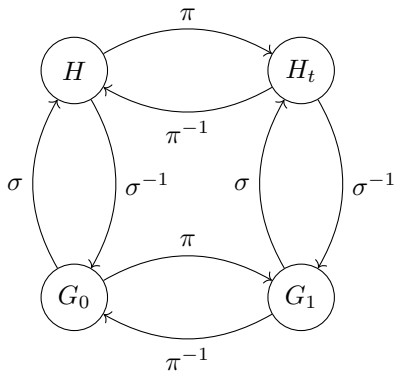
An interactive proof system is zero-knowledge if whatever can be efficiently computed after interacting with $\mathcal{P}$ on input $x \in L$ can also be computed given only x. This should be true even when $\mathcal{P}$ is interacting with a malicious verifier.

Again, this is most easily demonstrated with an example. Let us return to Graph Isomorphism, and show that we can prove the input graphs G_0, G_1 are isomorphic without revealing the isomorphism.



Proof method

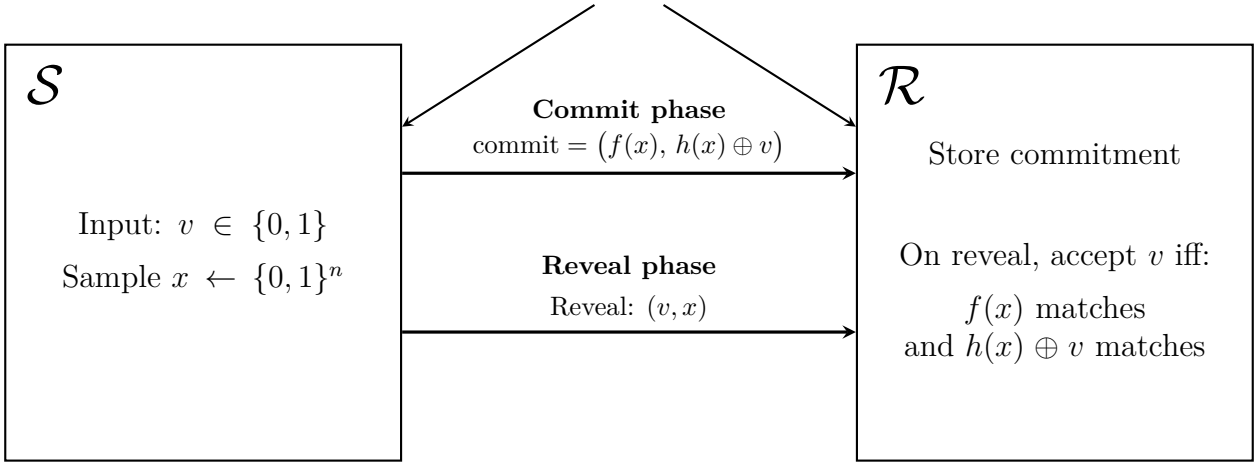
Consider at the same time that we have the following graphs, allowing us to demonstrate the isomorphism π between G_0 and G_1, without ever revealing it:



18 Commitments

An example bit commitment scheme:
Given a PRG $G : |G(s)| = 3 \cdot |s|$, and a hard-core predicate $h : \{0,1\}^* \to \{0,1\}$, the sender is given $v \in \{0,1\}$ as input.

Public parameters: OWF f, hard-core predicate h

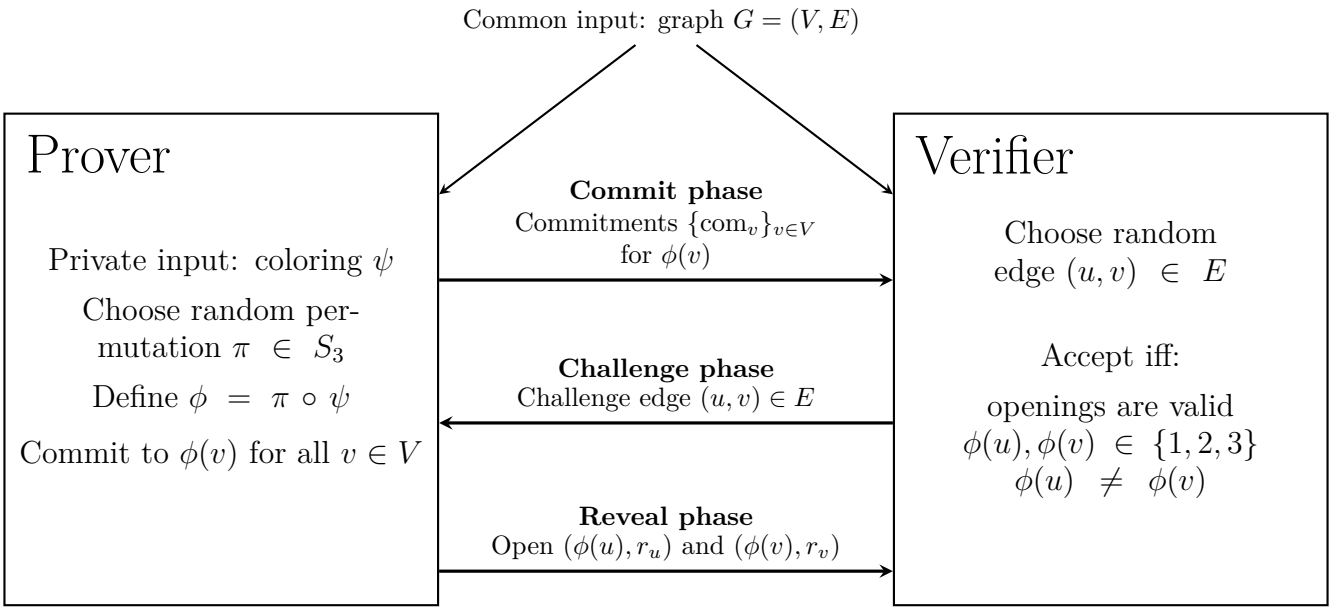


Bit-commitment scheme (OWF-based)

As we can see, in the commitment (very *very* informal), $f(x)$ is functioning as a signature to verify the value of x, and $h(x) \oplus v$ is function as a signature to verify the value of v.

This can be extended to coin flips over the telephone (for example), by having Alice commit to her result, Bob respond with his result, and Alice then reveal her result. This way, neither Alice, nor Bob can change their results according to what the other said.

19 ZKP for G3C with Commitments



One round of the G3C zero-knowledge protocol

20 Cryptography Primitives