

<u>Outline:</u> - Commitments - Motivation - Definition - Example: Pedersen Commitments Random Oracle Model - Simple Commitment Scheme from Hash Function - Intuition - Formalization - Example : Simple PRF - Intuition
- Formalization
- Example: Simple PRF
Commitment Scheme Motivating Example: - Alice and Bob want to "flip a coin" over the phone $4t$ tempt #1: Alice Bob 1.Alice flips coin and gets b 2. Alice sends b to Bob Simple PRF
me Motivating E
Bob want to"
hone
tice sends b to Bab Alice outputs b Bob outputs b Problem : What if Bob doesn't trust Alice?

Attempt #2: Alice Bob 1. Alice flips coin and gets ba 2.Alice locks ba in box Attempt #
Alice flips L
1. Alice flips L
2. Alice locks ba 3.Alice sends locked box to Bob - 4. Bob flips coin and gets b_s $\frac{a}{2}$
5. Bob sends b _B to Alice 6. Alice sends locked

box to Bob

a

Bob sends b₈ to Alice

Alice sends key to Bob

b₈

Bob out;

Bob out;

Bob out;

Bob out;

Bob out;

Bob out;

Bob out; to Bob Alice outputs $b_A \oplus b_B$ Bob outputs $b_A \oplus b_B$ Alice outputs $b_A \oplus b_B$
Claim: $b_A \oplus b_B$ is random if at least I party is honest Why? 1) If Alice honest, ba is uniformly random . Bob knows nothing about ba when choosing be \Rightarrow ba and by are independent
 \Rightarrow ba \oplus by is also uniformly random.
 \Rightarrow If Bob honest, by is uniformly random. Alice can't \Rightarrow $\frac{1}{2}$ $\frac{1}{$ $changebas \geq ba$ $changebas \geq ba$ and b_B are independent $\Rightarrow b_A \oplus b_B$ uniformly random

Commitment Schemes

Intuition : Cryptographic opaque locked box

Formally : ^a pair of algorithms T⁼ (Setup, Commit) - Setup(1^2) = pp takes security parameter as input and $outputs$ public parameters
- $Commit (95 M)^3$ $Countputs$ public par
Commit (pp, m, r) >c takes public parameters as cuput, a message

m from the message space M, and

randomness r from randomness space R

and outputs a commitment in commitment

space C

Two Key Properties and outputs a commitment in commitment space C Kes public parantes public parantes of the
domness of the
domness of the
parantes a control
parantes hide
can't see the
lumit see the
stical: statistica
tational: no ef-

space
Commitments hide the message
Hiding: commitments hide the message Um, m'e M. $m \in M$ U "you can't see through the Box"
m,m'e M
{ Commit (m,r): r&R3 = { Commit (m',r): r&R3

- perfect : distributions same - statistical : statistical distance negligible computational: no efficient distinguisher

Binding: no efficient adversary can produce m, m', r - perfect: distributions same
- statistical: statistical distance negligible
- computational: no efficient distinguisher
Binding: no efficient adversary can produce m,m', r,
binding: no efficient adversary can produce m,m' r no errorent daversary can produce m, m, r, r
such that Commit (m,r) = Commit (m',r') "Only one value locked in the box" 4 PPT $\ddot{4}$ Pr $[Commit (m, r) = Commit (m', r') : (m, r, m', r') \leftarrow A (pp)$ $\frac{1}{2}$ negl(λ) neg

computational

Vedersen Commitments Construction G: a group of prime order p generated by g We assume the discrete log problem is hard in G:
Logiven (G, p, g, gx) for a uniformly random
 $x \stackrel{\text{d}}{=} Z_p$, it is infeasible to compute x Discrete-Log Security Game (formal) Adversary A Challenger $-$ x wins when $x = x^3$ D-log assumption: all PPT adversaries $C = G$ $M = \mathbb{Z}_p$, $R = \mathbb{Z}_p$, Commitment Scheme:
Setup (1):
- sample ht G, p, g, h) Commit $((6, p, q, h), m)$
- $r \stackrel{a}{\leq} Z_p^p$ and

Made with Goodnote

Analusisl & Commit (m,r) :R⁼ gmh : Pf . For any $m \in \mathcal{H}$, consider the distribution $\begin{array}{r} \n\text{For any } m \in \mathbb{Z}, \text{ consider the most } n \in \mathbb{Z} \n\\ \n\text{For any } m \in \mathbb{Z}, \text{ for all } m \in \mathbb{Z} \n\\ \n\text{For any } m \in \mathbb{Z}, \text{ for all } m \in \mathbb{Z}, \text{ and } m \in \mathbb{Z} \n\\ \n\text{For any } m \in \mathbb{Z}, \text{ and } m \in \mathbb{Z}, \text{ and } m \in \mathbb{Z}. \n\end{array}$ r is unit rand \Rightarrow h' is unit rand \Rightarrow g"h' is unit rand.
 \Rightarrow distribution is independent of m 2) Computationally Binding
To prove binding, we show d-log hardness => binding Binding
To prove binding, we show d-log hardness => binding prove binding we show d-log hardness => binding.
S. We assume we have an bolversary 4
that can break binding of Pedersen w/ non-negl Pf. We assume we have an dot probability ^p . We then show that we can use A to build adversary ^B that wins D-log game. B simultaneously plays the role of adversary in the D-log D simultaneously plays the role of adversary in the D-le
game and challenger in the Pedersen binding game
life Advice: to break d-log, get two different representations of a group element $g^{\mathsf{m'sxv'}}$ For example : m' in m'' = g^m (g^x)^r = g^m (g^x)^r = g^{m+xr} = g^{m-xr} = gm+xr = gm +xr = gm + $g^m h^r = g^m h^r$
 $\Rightarrow m+xr = m'$
 $g^m h^r = g^m h^r$
 $\Rightarrow m+xr = m'$
 (g, p, g, h)
 h, m', r, r' $\frac{m-m}{r-r}$ $x = \frac{m_1 + n_2}{r^2 - r^2}$

curity Game Pf.
 $\frac{m_1}{r}$ D.
 $\frac{m_2}{r}$ and $\frac{m_1}{r}$
 $\frac{m_1}{r}$ and $\frac{m_2}{r^2 - r}$
 $\frac{m_1}{r^2 - r}$ $\frac{m_2}{r^2 - r}$
 $\frac{m_1}{r}$
 $\frac{m_2}{r^2 - r}$ $\frac{m_1}{r^2}$ Pedersen Binding Security Game Pf dersen Binding Security Game Pf.
A ^{(Binding})
G, p, g, h
C, p, g, h
C, p, g, h $\frac{Cha||e}{\sqrt[3]{7}}$ $\frac{3}{x}$ M ag' m, m, r $\frac{1}{\sqrt{1}}$ $gmh = g''h''$ $r' = g'''h''$
 $\neq m$
 $= \frac{m-m'}{r'-r}$ usewice! \overrightarrow{x} $Pr[x' = x] = p$ $Pr[x' = x] = p$ which is not negligible!

Pedersen commitments are homomorphic Commit (m, r) . Commit $(m', r') = q^m h^r q^{m'} h^{r'}$ $=$ $\begin{array}{c} 0 \\ 0 \end{array}$ (m+m', r+r')

... but are there simpler commitment schemes?

YES!!

Hash-based commitments !

... but how to prove security?

We need the RANDOM ORACLE MODEL commitments!
now to prove security?
2ANDOM ORACLE MOD!

Random Oracle Model
Reasoning about security of hash
instead of reasoning about he
function H as a random fun
H : $x - y$ defined by
agrees w/ common intuition
Pervasive in cruptograph Random Oracle Model * controversia) Reasoning about security of hash functions is hard => instead of reasoning about hash directly, treat hash function ^H as ^a random function $H: \chi \rightarrow$ $-$ function H as a random function
H: $x \rightarrow y$ defined by $H(x) \mapsto a$ random element of y
-> agrees w/ common intuition for hash functions
-> pervasive in cryptographic implementations pervasive in cryptographic implementations Consider simple commitment with hash H modeled as random function.
Commit (m,r):= H (m,r) $Commit (m,r) = H(m,r)$ hiding: H's output is uniformly random
- binding: breaking binding requires finding
(m,r) = (m',r') such that Hlm,r) = Hlm',r') rmly roundom
requires finding
Hlm,r) = Hlm',r'), a collision, which is hard for ^a random function Q: Is H(m) ^a commitment ? Or gm? No! m may have insufficient entropy Elegant but some questions : * why ^a "model" not an assumption ? * how can we formalize this? Why ^a model ? Cryptography is lepistemologically) part of mathematics We model the world and prove theorems within the ne mou
model. mode).
La Our proofs so far have been in the standard model · weak assumptions ↳ Now we'll see the random oracle model · ^a stronger assumption : "all parties have Oracle access to ^H , ^a random function, sampled at $stat-up"$ Weakness: in our implementations, we do not
sample $H \Rightarrow the$ mode) is a LIE ? tronger assumption: "all parties have
ess to H, a random function, sample.
wt-up"
: in our implementations, we do no
sample $H \Rightarrow$ the model is a LIE!

Madewrith Goodnotestall models are wrong; some are useful"

How to formalize ? $16w$ to formalize:
- let $H: X \rightarrow Y$ be a function (i) $\begin{array}{lll} \text{low} & \text{to} & \text{for} \text{malize} \\ \text{let} & \text{H: } \chi \to \chi \text{ be a function } l \text{ te-the random oracle} \end{array}$ all parties have access to an oracle that samples ^H Oracle
 x^3 H = Funs [x, y]

x

x

x

x

x

x

x

x Adversary Challenger ↳ in security game proof, adversary sends RO queries to Challenger Security Game Pf in RO Model
Adv A Challenger 000 suive proof, adverse
challenger
If in ROModel
game query Cha RO query, ame respons.
RO query, $\overline{\mathcal{O}}$ response

↳ Challenger's responses must be to ^a random function 2 2 to a random function
Lo Ex: for each adversary query H(m)
C sets H(m) 2 4 and C sets Hlm) C and
C sets Hlm) C y and

PRF Proof in RO Model

PRF Security Game for F $KxM \rightarrow Y$ that uses R O. H Adversary - RF Proof in RO Model
RF Security Game for F: KxM -24 th
Haversary AO (hallenger (b)
f @ m? If b=0, k=+ $\frac{1}{\begin{array}{r} \text{node} \\ \text{F: } k \times k \rightarrow 2 \\ \text{In the user } k \\ \text{if } b = 0, k = 1 \\ \text{else } k = 1 \\ \end{array}}$ $f @ m?$ If $b=0$, $k=K$, $f \in F(k,.)$
 $f @ m?$ if $b=0$, $k=K$, $f \in F(k,.)$ 64
 64
 $+ 24$ $Game$ for 1
 $\frac{1}{2}$
 $\frac{1}{2}$ $\begin{array}{c}\n\uparrow @m? \\
\uparrow @m? \\
\hline\n\uparrow @m? \\
\hline\n\downarrow @m? \\
\hline\n\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \end{array}$ H & ^m? $\begin{array}{c} f(m) \ \hline \downarrow 0 m? \ \hline \uparrow \downarrow 0 m? \ \hline \end{array}$ $\begin{array}{c|c}\n\hline\nf(m) & & \\
\hline\nH(m) & & \\
\hline\n\end{array}$ $\vert b \vert$ Let W_b be the event that A outputs I when $b = 0, 1$ As advantage with respect to F is PRFadv [A, F] :=

 $PRE[W_0] - PrE[W_1]]$. A PRF F is secure if for all efficient adversaries, PRFadv [A, F] is negligible

Now let's use this definition to prove security of ^a specific hash-based PRF !

The PRF: $f(k, x) = H(x)^k$, $H: x \rightarrow G$ Claim : Secure in RO model assuming DDH

Decisional Diffie-Helman (DDH) Assumption : for group of order a with generator g. nal Diffie-Helman (DDH) Assumption:
or group of order a with generator g:
{ (g^x, g³, g^xr) : x, y = Z g} = c { (g^x, g³, g³⁾ ×, y, z = Z }

 \downarrow \downarrow \downarrow "DDH triple" "random triple"

DDH Security Game

 t

1 Security Game
Adversary B Challenger (b)
X, y, z et (hallenger(b) <u>hallenger (b)</u>
X, y, z c⁺ Zq $x, y, z \leq z$
if $b = 0$, $x = g^x$, $y = g^y$, $z = g^{xy}$ X, Y, Z else $X =$ g^{x} , $y = g^{y}$, $z = g^{z}$ x, y, z
 \overline{y}
 \overline{z}
 \overline{z}
 \overline{z}
 \overline{z}
 \overline{z} \overline{D}

Let W_p be the event B outputs I for $p = 0$). B's advantage in solving DDH for ^G is DDHadv[B, G]: ⁼ /Pr [Wo] - Pr[Wil Eine say the DDH assumption holds for G if for all efficient adversaries ^B the quantity DDHadv [B, G] s negligible

We will prove that if there exists an adversary A who breaks our PRF, then we can build an adversary B who uses A to break DDH, Just as before, B simultaneously plays the role of challenger in the PRF security ses in to preak buil, was as becore, in
plays the role of challenger in the PRF will prove that if there exists an adversary A who
aks our PRF, then we can build an adversary B who
s A to break DDH, Just as before, B simultaneously
us the role of challenger in the PRF security
and adversary in the DD

RO ame
A $x, y, z \stackrel{\epsilon}{\leq} Z$ $x, y, z \in \mathbb{Z}^q$
if b=0, x=g, y=g, z=g,y V if $b=0, \frac{x}{3}, \frac{y}{2}=\frac{y}{3}$
 $X, Y, Z = \frac{y}{3}, \frac{y}{3}=\frac{y}{3}$
 $X, Y, Z = e^{x} \times e^{x}$
 $e^{x} \times e^{x} \times e^{x}$
 $Y = e^{x} \times e^{x} \times e^{x}$ $9, 4^{\circ}$ $\begin{array}{c} \n\epsilon & 0,2 \\
\epsilon & 2\n\end{array}$ Z f & ^m? $f(m)$
H Qm ? Magically determine $H(m)$ if interacting with Magically determine
If interacting with
PRF or [random](https://goodnotes.com/) function b ,

*^B has to convince ^A it is interacting with ^a "real" PRF challenger - > its query answers must be indistinguishable from aulswers from
"real" PRF challenger. "real" PRF challenger -> its query arswers Trick : ^B doesn't use RO-it just pretends to ! 3 has to convince A it is
real" PRF challenger -> is
rust be indistinguishable
real" PRF challenger.
Trick: B doesn't use RO-it j
PDH Adversary B(pp, X, Y, 2)
T-> B main tains map H
simulated RO value: $\frac{1}{\sqrt{2}}$ B maintains map H to keep track of simulated RO values for consistency

- To answer H @ m querres :

- If m & H : $rac{6}{x}$
 $rac{6}{x}$
 $rac{1}{x}$ indistinguishable from $-\alpha \stackrel{\text{max}}{\leftarrow}$
- $\alpha \stackrel{\text{max}}{\leftarrow}$
- Set $H(\alpha) \leftarrow (\chi^{\alpha} \alpha)$ $x \in \mathbb{Z}$
Set $\text{H}(m) \in (\chi^{\alpha}, \alpha)$ - Send H(m)[O] Les To answer f @ m queries : unswer
If m 4 H:
- a 4 Z; $- \alpha \stackrel{4}{\leq} \frac{H}{Z_q}$ Set $H(m) \leftarrow (X^{\alpha}, \alpha)$ - $- \alpha \leq \mathbb{Z}_{q}$
- Set H(m)
- Q H(m)[1]
- Send Z^{α} $\alpha \leftarrow H(m)\Sigma 1$
Send $Z^{\alpha} \leftarrow$ if $b=1$, then this is PRF where y is secret key ! Note: $i\hat{f}$ b=0 $if b = 1,$ and $Z^{\alpha} \leftarrow$ if $b=1$, then this is PKF where

y is secret key!
 $Z^{\alpha} = g^{x}y^{\alpha} = g^{x\alpha}y = X^{\alpha}y = H(m)y^{\alpha}$ the PRF!
 $Z^{\alpha} = g^{z\alpha} = (g^{\alpha})^{z}$ indistinguishable from Ergo , guessing PRF vs. random is equivalent to guessing DDH triple vs. random triple!
=> B and A have the same advantage! \Rightarrow if A can break PRF security, then B can break DDH assumption

Iboughts and Comments on RO Model

- a heuristic model that seems to work well in reality and gives simpler/faster schemes than we have in the standard model
- in applications , we replace RO with a specific nash function > pretty pervasive in implemented crypto
- there are some (contrived) schemes that are secure in RO model but insecure in standard model no matter what hash function is used
- Some people especially don't like this "dirty trick" some people especially don't like this dirty trick
of programming an RO-how is it connected to reality ??

 we lat Stanford) tend to be RO-friendly

On Instantiation - do not use Merkle-Damgard hash like SHA256 SHA3 (sponge-based) or -SHA2, carefully padded