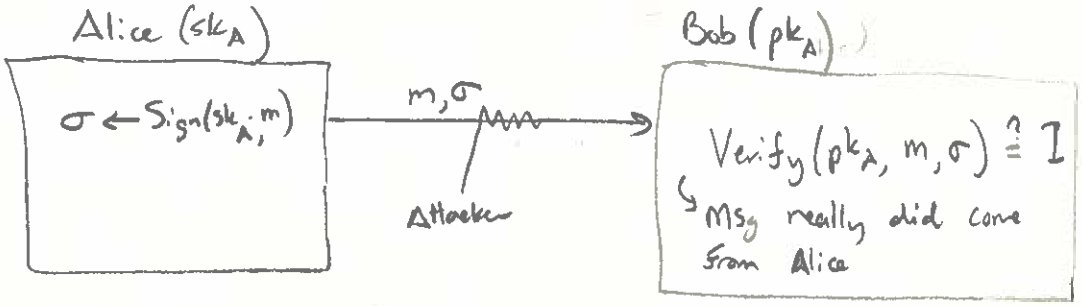
PROBLEM SET OUT!

[Write David's timeline on board] → ZK next week

In this lecture, we will cover

* Digital signatures (recap)
* TDPs (recap)
* Random-oracle model
  ↳ prove security of RSA-FDH

Recap: Digital Signatures
$$\left\{\begin{array}{l} Gen(1^\lambda) \to \{sk, pk\} \\ Sign(sk, m) \to \sigma \quad \text{for } m \in \mathcal{M}_\lambda \\ Verify(pk, m, \sigma) \to \{0,1\} \end{array}\right\} \text{efficient}$$

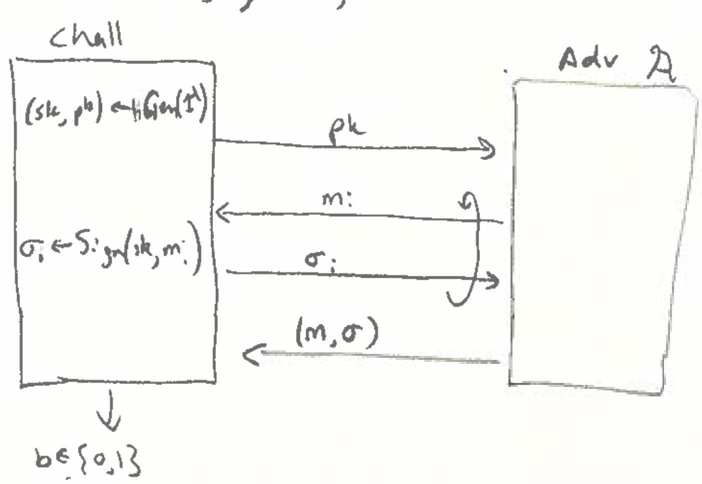


To be useful must be

1) Correct: For all $m \in \mathcal{M}$, $(sk, pk) \xleftarrow{R} Gen(1^\lambda)$
$$Pr\left[Verify(pk, m, Sign(sk, m)) = 1\right] = 1$$
→ Honest signer accepts honest signature

2) Secure: "Should be hard to cook valid σs w/o sk"

Existential unforgeability under chosen-msg attack (EUF-CMA)



We say Adv wins if
1) $m \notin \{m_1, \ldots, m_q\}$ – m is new
2) $Verify(pk, m, \sigma) = 1$ – sig σ is valid

$SIGAdv[\mathcal{A}, S] = Pr[\mathcal{A} \text{ wins game}]$ → should be negl in λ

Recap: trapdoor one-way permutation → Intuition: easy to go forward, hard to invert.
BUT ∃ a sk that allows eff. inversion

Three algs ~~ ~~ defined over $\mathcal{X}_\lambda$,

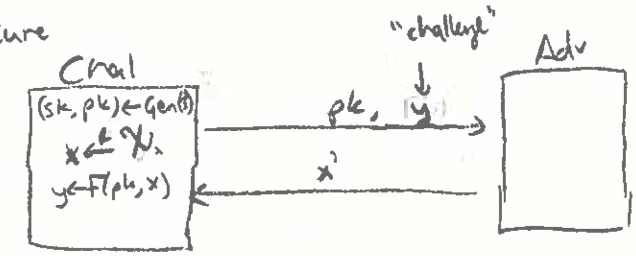
$$\left.\begin{array}{l} Gen(1^\lambda) \to (sk, pk) \\ F(pk, \overset{\in \mathcal{X}}{x}) \to y \in \mathcal{X} \\ F^{-1}(sk, \overset{\in \mathcal{Y}}{y}) \to x \in \mathcal{X} \end{array}\right\} \text{efficient algs}$$

To be useful:

1) Correctness for all $\lambda \in \mathbb{N}$, $x \in \mathcal{X}_\lambda$ $(sk, pk) \xleftarrow{R} Gen(1^\lambda)$

$$\Pr[F^{-1}(sk, F(pk, x)) = x] = 1$$

2) Secure



Say adv wins if $x' = x$.

$OWAdv[A, \mathcal{T}] = \Pr[x' = x]$ should be negl in λ.

⟹ Can build TDPs from RSA (and not really anything else) factoring (as far as we know)

↳ Essentially abstracts away details of RSA (primes, etc...)

Idea: Build Digital Sigs from TDPs.

Broken idea: use TDP directly to sign.
If TDP $F: \mathcal{X} \to \mathcal{X}$, then msg space is $\mathcal{X}$, sig space is $\mathcal{X}$

$\text{Gen}_{\text{Sig}}(1^\lambda) \to$ output $(pk, sk) \xleftarrow{R} \text{Gen}(1^\lambda)$

$\text{Sign}(sk, m) \to$ output $\sigma \leftarrow F^{-1}(sk, m)$

$$\text{Verify}(pk, m, \sigma) = \begin{cases} 1 & \text{if } F(pk, \sigma) \stackrel{?}{=} m \\ 0 & \text{o.w.} \end{cases}$$

When TDP is RSA, this is known as "textbook RSA signatures"

Problem: for any σ^*, can compute $m^* \leftarrow F(pk, \sigma^*)$

Now (m^*, σ^*) is a valid signature

↳ Anyone can forge signatures given only verifier's public key!

↳ Note that the value of m^* is not really under the adversary's control (essentially are forging a sig on a randomish msg).

↳ Still bad! → Violates our security defn

Intuition: $F(pk, \cdot)$ is only hard to invert on a <u>random</u> input
Here, adv gets to choose input to $F(pk, \cdot)$

Signatures in ROM

Turns out, it's very hard to construct practical signatures from standard/simple assumptions (e.g. RSA/TDP)

Two alternatives

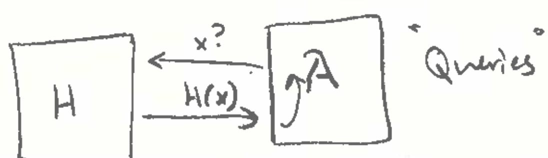
1) Use stronger assumptions: "Strong RSA"
2) Use a different model of computation: ("Random-oracle model")

The R.O.M. is amazingly useful

↳ the first tool of choice for constructing & analyzing security of "practical" cryptosystems

Idea: Model a cryptographic hash fn $H: \{0,1\}^\lambda \to \{0,1\}^\lambda$ as a truly random function, to which all parties have "oracle" access

exponentially large — too big to write down



→ In practice: instantiate H w/ SHA256 (or similar) ← Be careful!

→ In this model, easy to construct good sig schemes from TDPs!

We measure the running time of sig forger $\mathcal{A}$ by

1) # of signing queries it makes
2) # of random oracle queries it makes

Caveat: ROM is "too good to be true"

⟹ Famous result: There exists sig scheme S secure in the ROM s.t. for every hash fn H, S is insecure when instantiated w/ H

[Canetti, Goldreich, Halevi JACM'04]

↳ R.O. model doesn't say anything about what happens when you replace R.O. w/ real fn!

↳ And yet! It's amazingly useful way to analyze hash fns

Full-Domain Hash S_{FDH} using TDP $T = (Gen_{TDP}, F, F^{-1})$

$\triangle$ non-broken sig scheme from TDPs, hash fn $H : \mathcal{M} \to \mathcal{X}$

model as random oracle

↳ $Gen_{sig}()$: output $(sk, pk) \xleftarrow{R} Gen_{TDP}()$

$Sign(sk, m \in \mathcal{M})$: output $F^{-1}(sk, H(m))$

$Verify(pk, m, \sigma)$: output $\begin{cases} 1 & \text{if } H(m) = F(pk, \sigma) \\ 0 & \text{o.w.} \end{cases}$

"Hash and sign sig scheme"

Thm Let $\mathcal{A}$ be an eff adv attacking S_{FDH} that

1) Uses Q_s signing queries
2) Uses Q_{RO} r.o. queries

then there exists an eff $\mathcal{B}$ s.t.

$$SIG^{RO}Adv[\mathcal{A}, S_{FDH}] \le (Q_{RO} + Q_s + 1)\, OWAdv[\mathcal{B}, T].$$

Let's pause to see what this means.

"East-coast view": $Q_{RO}, Q_s \in poly(\lambda)$

$OWAdv[\mathcal{B}, T] \in negl(\lambda)$

implies $SIG^{RO}Adv[\mathcal{A}, S_{FDH}] \in negl(\lambda)$

Asymptotically secure....

"West-coast view" $Q_{RO} \approx 2^{40}$ ← How many times can adv eval SHA256?

$Q_s \approx 2^{??}$ ← How many chosen sigs can adv get?

$\Rightarrow$ If $\exists$ adv that forges w.p. ε, then $\exists$ adv that inverts OWF w.p. $\varepsilon' \approx \varepsilon / 2^{40}$.

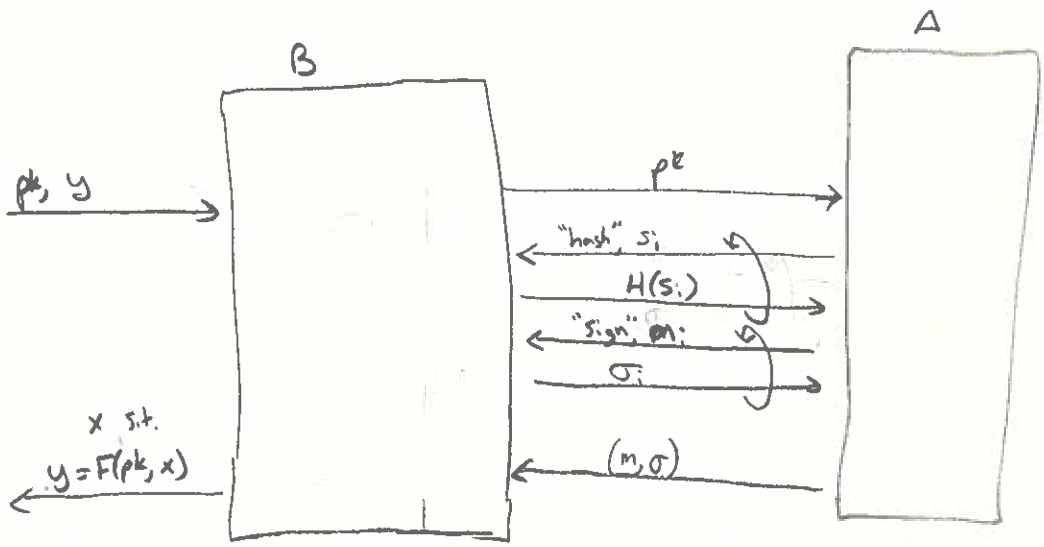
"Loose reduction"

See Boneh-Shoup (Sec 13.5) for info on scheme w/ tight reduction $\varepsilon' \approx \varepsilon$

(Also Bellare & Rogaway)

Security Pf Idea (Boneh-Shoup Thm 13.3)

Must construct B



→ Must use forger A to invert TDP.

→ Must obey the "API" of A:

* responses of hash queries should be rand-looking strings
* response of sig queries " " valid sigs on m_i w/ pk

Assume that: * A queries r.o. on every msg issued in sig query / forgery
* A makes distinct queries

} increases # of R.O. queries by $Q_s + 1$

Alg B

- Guess which R.O. query is forgery (index i^*)
- When A makes ith R.O. query
  - if $i = i^*$, respond w/ y (TDP challenge)
  - else, choose $x_i \xleftarrow{\$} \mathcal{X}$, respond w/ $y_i \leftarrow F(pk, x_i)$
- When A makes sig query on msg m s.t. $H(m) = y_i$
  - ↳ Can always respond w/ x_i
- If our guess is correct → A's forgery gives us (m, σ) s.t.

$$H(m) = y = F(pk, \sigma)$$

↖ σ is inverse of y under F

What happens if we guess wrong? $\Rightarrow$ Can't invert!

$$\Pr[\text{B guesses correctly}] = \frac{1}{\#\text{RO queries}} \geq \frac{1}{Q_s + Q_{ro} + 1}$$

So $\mathrm{OWFAdv}[B, T] \geq \dfrac{\mathrm{SIG}^{ro}\mathrm{Adv}[A, S_{FDH}]}{Q_s + Q_{ro} + 1}$

This proves the thm.

Why did we need the R.O. model?

↳ let us "stick in" the TDP challenge to the adv