Logistics
- Problem set due now
- Scribe for lecture
- Feedback summary

- Write bigger, more slowly
- Relevant modern readings – historical papers are hard to read!
- Big picture
- History / color commentary
- Bitcoin (probably won't do Bitcoin class)
- Come to OHH if you want clarification

"The Big Picture"
Review of RSA

Hard Problems
- Factoring: Given \( N= pq \), produce \( (p,q) \)
- RSA-e: Given \( N, a \in \mathbb{Z}^* \) produce \( x \) s.t. \( x^e = a \pmod{N} \)
  "Find the \( e \)-th root \( \pmod{N} \)
- Strong RSA: Given \( N, a \in \mathbb{Z}^* \) produce \( (x,e) \) s.t. \( x^e = a \pmod{N} \)
  with \( e \neq \pm 1 \)
  "Find any \( e \)-th root \( \pmod{N} \)

\[ \text{FACTORIZATION} \geq \text{RSA} \geq \text{Strong RSA} \]

Random Self-Reduction
- If RSA-e is hard for any \( a \in \mathbb{Z}_N^* \), it is hard for almost all \( a \in \mathbb{Z}_N^* \).

Rabin TDOUF
- \( f(x) = x^2 \pmod{N} \)
- We argued that inverting \( f(x) \) for \( x \in \mathbb{Z}_N^* \) is as hard as factoring \( N \).
- Can build PRE from Rabin's function (also signatures)
  \( \Rightarrow \) we didn't explain how (\text{\$255})
  \( \Rightarrow \) Taking square roots \( \pmod{N} \) is as hard as factoring \( N \)!
Hash-and-Sign Signatures

\[ \text{Sign}(sk, m) = H(m)^d \mod N \]

\[ \text{Verify}(uk, m, \sigma) = \left\{ \sigma^e = H(m) \right\} \mod N \]

\( \Rightarrow \) Without the hash, it's broken!

Threshold RSA Signatures

1. Split the signing key into \( p \) parts
2. Need all \( p \) parts to sign

Blind Signatures

- Can sign a message without knowing what it is
- E.g. anonymous survey w/ extra credit

\[ \text{Student} \hspace{1cm} \text{US} \]

\[ \begin{array}{c}
\text{Anonymously receive token}
\end{array} \]

\[ \begin{array}{c}
r \cdot H("henrycg")
\end{array} \]

\[ \begin{array}{c}
\text{Claim extra credit}
\end{array} \]

\[ \begin{array}{c}
\Phi - H("henrycg")
\end{array} \]

RSA Accumulation

- Shorter Merkle tree
In this lecture
1) Elliptic curves: what & why?
2) Hard problems related to EC.
3) Application: PRF from DH
4) Pairings & applications

Elliptic Curves

First, why: x an RSA signature is an element of $\mathbb{Z}_N^*$

- For 128-bit security, need $\geq 3072$-bit modulus
- Best attack cost $\approx 2^{118}$ work

Reason: best alg factors a $n$-bit int in time $\approx 2^{3.37 n^{1/3} (\log n)^{2/3}}$

If you want to choose $n$ s.t.

\[ 2^{3.37 n^{1/3} (\log n)^{2/3}} \geq \frac{1}{\varepsilon} \]

\[ \Rightarrow n \text{ grows like } 2^{3.37} \]

In contrast, these fancy attacks do not apply to EC systems

- Best attack on $n$-bit keys: $2^{n/2}$ time

\[ \Rightarrow n \text{ grows like } 2n \]

\[ \Rightarrow \text{For 128-bit security, 256-bit key} \]

Signature are nearly as short (10x shorter!)

Proposed in 1980s by Koblitz & Miller

- Certicom (Canadian company) pushed ECC study in 80s
- RSA Corp lobbied heavily against ECC

2005 - NSA pushed industry to switch to ECC

... took until 2011 for Google to make ECC default in TLS

Why took so long?

- Suspicions of NSA backdoors - magic constants!
- Later proved invalid

- 'Mark is harder - RSA is relatively easy, but a program to see
- Lists of subtle special-case attacks
  - # points in $E/F_p = p - \text{anomalies}
  - ECC = $\text{Diego Fp}$
  - ECC over $F_p$ over in large
  - Human: invalid curve

(continued on back)
EC: What Changed

1988 - NFS (Pollard) → Bigger RSA modulus

2004 - IBE from Pairings (DAN)
   - Positive applications
   - Lots of academics have a stake in ECC success

2010s - TLS becomes widespread
   - Saves on bandwidth, key resource
Elliptic Curves

A group is \((G, \ast)\)

- set of elements
- binary operation

1. Closure
2. Associativity
3. Identity
4. Inverse

In "Classic DH"

\[ G = \{\text{integers mod prime } p\} \]
* = "multiplication mod p"

In \(E \subset \mathbb{DH}\)

\[ G = \{\text{set of } (x,y) \text{ points on curve } E\} \]
* = "addition of EC points"

Elliptic Curve

- Work mod prime \((p \times 256 \text{ bits})\)

\[ y^2 = x^3 + Ax + B \quad (\text{mod } p) \]

- Over \( \mathbb{R} \)

\[ y^2 = x^3 - 5x + 4 \]

- Why EC?

\[ G = \left\{(x,y) \in \mathbb{Z}_p^2 \mid y^2 = x^3 - 5x + 4 \text{ mod } p\right\} \]

\(U\) \{point at infinity\}

What about group \(\ast\) ?

\[ 3 \ast g_1 = \text{draw line & reflect} \]

\[ 3 \ast g_2 = \text{draw tangent & reflect} \]

\[ 3 \ast \mathcal{O} = \text{do nothing} \]

\(\Rightarrow\) Lots of deep theory here

* 4 properties of group hold.

* Point compression:

- Every \(x\), only \(\leq 2\) \((x,y)\) points on \(E_{2,0}\)
- Send \((x, y)\) instead of \((x, y)\)
EC Notation

$g = \text{point on curve}$

$g^2 = g \cdot g = \text{point composed with itself}$

$g^3 = g \cdot (g^2)$

$g^a = g \cdot (g^a)$

$g^a = \text{point composed with itself a times}$

"generator"

If there are $q$ points on $E$, $q$ prime $3$ point $g$ st.

$G = \{g, g^2, g^3, \ldots, g^q, g^{q+1}\}$

$G$ works like our normal DH group!

Order is $q$.

Sometimes you'll see the confusing notation

$G = \{P, 2P, 3P, \ldots, (q-1)P, qP\}$

Computational Problems in $G$:

For EC $G$, generator $g$ of order $q$.

Discrete Log: $a \in \mathbb{Z}_q$

Given $(g, g^a)$ produce $a$.

CDH: $a, b \in \mathbb{Z}_q$

Given $(g, g^a, g^b)$ produce $g^{ab}$.

DDH: $a, b, c \in \mathbb{Z}_q$

Given $(g, g^a, g^b)$ or $(g, g^b, g^c)$ identify which you've been given.
EC Hard Problems

Factoring $\Rightarrow$ RSA $\Rightarrow$ Strong RSA

Disj. $\Rightarrow$ CDH $\Rightarrow$ DDH

$\Rightarrow$ Best algorithm for DDH is "brute force" dlog
when $p = n$ bits, $2^{n/2}$ time.

For 128-bit security, need a 256-bit prime

Mysteriously in Aug 2015, NSA changed to require 384-bit prime

Do you have a better alg for ECDlog? 😊

---------

DDH is useful (see Dan’s paper)

- It’s a qualitatively different type of assumption
- Search: factor $N$, recover $e$, then run $\text{find dlog}$
- Decision: distinguish these distributions

$\{ (g, g^a, g^b) : g \in \mathbb{G}_1 \}$ vs $\{ (g, g^a, g^b) : g \in \mathbb{G}_2 \}$

- For building crypto, decision is useful!
- DDH has a randomized self-reduction!
- Either easy everywhere or hard almost everywhere
- DDH is easy in $\mathbb{Z}_p^*$!
Application: PRG from DDH

Recall PRG

\[ f : \mathbb{Z} \rightarrow \{0,1\}^n \quad (\text{s.t. } n \geq k \cdot 1^k) \]

\( \{ k \in \mathbb{Z} : f(k) \} \approx \{ z \in \{0,1\}^n : z \} \)

Simple PRG

Fix \( g, g^a \) (for random \( a \in \mathbb{Z}_q \))

\[ f : \mathbb{Z}_q \rightarrow G \]

\[ f_{g^a}(k) = (g^k, g^{ak}) \]

Security is immediate under DDH:

\[ \{ (g^k, g^{ak}) \} \approx \{ (g^k, g^a) \} \] Random over \( G \)!

By DDH

N.B. CDH/Dlog not enough!
Application: PRF from DDH

Recall a PRF $f: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{Y}$

$\{ k \in \mathcal{X} : \delta(k, \cdot) \} \approx \{ F \in \text{Fun}[\mathcal{X}, \mathcal{Y}] : \delta(F) \}$

Naor-Reingold PRF

Omer

$\mathcal{X} = \mathbb{Z}_q^{n+1}$, $n+1$ field elements

$\mathcal{X} = \mathbb{Z}_q^n$

$\mathcal{Y} = \mathbb{Z}_q$

$\mathcal{Y} = G$

$\bar{k} = (k_0, k_1, \ldots, k_n) \in \mathbb{Z}_q^{n+1}$

$x = x_1, x_2, \ldots, x_n$

$s_k(x) = g^{k_0 T_{i=1}^n k_i} \in G$.

Then if DDH holds, $s$ is a secure PRF.