I. Roadmap

- Digital Signatures
  - Schnorr's Digital Signature
  - Fiat-Shamir Heuristic
  - Builds on Schnorr's ID Scheme
  - Builds on

II. Schnorr's ID Scheme

- Goal: Holder of secret key (Sk) should be able to convince others they hold the Sk without revealing info about it
  \[ (Pk, Sk) = (g^x, r) \]

- Setup:
  - Pick prime number q
  - We work in mod p, where \( p = q \cdot r + 1 \)
\[ G_r = \{ h^r \mod p, \ h \in \mathbb{Z}_p^* \} \]
\[ |G_r| = q \]

- Choose \( g \in \mathbb{Z}_p^* \) s.t. \( h^r \neq 1 \pmod{p} \)
- Let \( g = h^r \pmod{p} \)
Proof of Knowledge

- Holder picks secret key $x \in \mathbb{Z}_q$
  - publishes PK $g^x$

**Prover (P)**
- (knows $x$)

- $k \in \mathbb{Z}_q$

**Verifier (V)**
- (knows $g^x$)

- $r = g^k \pmod{p}$

P commits

- challenge

- response

V responds
- challenge

- V accepts iff

$$g^s = g^{k-xe} = \frac{g^k}{(g^x)^e} = \frac{r}{p^e}$$
2 properties:

1) Prover must know $x$ if he can answer most challenges in the form $\overleftarrow{e_i/s_i}$

Proof: Suppose $P$ responds to challenges $e_1$ and $e_2$ with $s_1$ and $s_2$.

$$g^{s_1} = \frac{r}{PK^{e_1}}, \quad g^{s_2} = \frac{r}{PK^{e_2}}$$

$$g^{s_1} \cdot PK^{e_1} = g^{s_2} \cdot PK^{e_2} = r$$

$$g^{s_1-s_2} = PK^{(e_2-e_1)}$$

$$g = g$$

$$x = \frac{s_1-s_2}{e_2-e_1}$$

$P$ must know $x$!
2) \( V \) gains no information about \( x \).

Key assumption: \( V \) is honest

(i.e. \( V \) picks \( e \) from \( \mathbb{Z}_q \) at random)

Proof: Verifier can generate a valid interactive transcript on their own WITHOUT knowledge of \( x \).

How?

\( V \) chooses \( e, s \) at random from \( \mathbb{Z}_q \)

\( V \) uses \( e, s \) to compute \( r \):

\[ r = g^s \cdot PK^e \]

\( V \)'s transcript = \((PK, r, e, s)\)

*The type of interaction seen in Schnorr's ID scheme is called Honest Verifier Zero Knowledge.*
III. Fiat-Shamir Heuristic

- Converts an interactive proof of knowledge into a digital signature

How to convert Schorr’s ID scheme?

- In challenge step, $V$ sends to $P$
  
  \[ e = H(m, r) \]
  
  * $H$ is CR hash function
  * Assuming ROM

- Scheme:

  \[
  \text{Sign}(SK, m) = (r, e, s) \]

  \[
  \text{Verify}(PK, m, (r, e, s)):
  \begin{align*}
  & \text{accept if ID scheme verifier accepts} \\
  & \text{accepts}
  \end{align*}
  \]
IV. Schnorr’s ID Scheme Example

- We’ll use $q = 5$ and $p = 11$
  - In practice, $q$ is 160 bits to avoid discrete log attacks.
    $p$ is selected to be 1024 bits to avoid birthday paradox attacks

Setup: $q = 5$ \quad p = 5 \cdot 2 + 1 = 11
\[ G_1 = \{ h^2 \mod 11, h \in \mathbb{Z}_{11}^* \} \]
\[ = \{ 1, 4, 9, 5, 33 \} \]

For generator $g$, we pick $h$ from $\mathbb{Z}_{11}^*$.
Say we pick $h = 9 \Rightarrow g = 9^2 (\mod 11)$
\[ g = 4 \]

Pok: User runs Gen to produce $(SK, PK)$ pair $(x, g^x)$.
Say we picked $x = 3$ randomly from $\mathbb{Z}_q$
\[ g^x = 4^3 (\mod 11) = 9 \]
Run 3-phase protocol:

\[ k \leftarrow \mathbb{Z}_q, \text{ say } P \text{ drew } k=2 \]

\[ r = 4^2 (\text{mod} \ 11) = 5 \]

\[ e = 1 \]

\[ s = 2 - 3 \cdot 1 (\text{mod} \ 2) = 4 \]

\[ e \leftarrow \mathbb{Z}_q, \text{ say } V \text{ drew } e=1 \]

Verifier’s check:

\[ g^s (\text{mod} \ p) = 4^{44} (\text{mod} \ 11) = 3 \]

\[ \frac{r}{P} (\text{mod} \ p) = \frac{5}{(9^4)^{-1} (\text{mod} \ 11) = 5 \cdot 5 (\text{mod} \ 11)} = 3 \]

\[ \Rightarrow P \text{ must know the discrete log of the public key } g^x = 9 \text{ in modulo 11.} \]
IV. Digital Signature Algorithm (DSA)

- Nearly identical to Schnorr's Signature Scheme.

Changes:

- In the commit phase,
  \[ r = g^k \pmod{p} \pmod{q} \]
  produces shorter signatures

- V's challenge to P is \( e = H(m) \) instead of \( H(m, r) \)

Full scheme details are in Lecture 12 notes.