Announcements

- Kryptos?

- Read all the presentation abstracts before next Monday

- Will post Exam 3 next Thursday
  Expect 25% to be essay similar to last semester

Group Presentations next week!

- Presentation should be 13-16 minutes long
- Rubric in onCourse
- Tutorial participation for next week is filling out short evaluation for other presentations in onCourse
- Practice, practice, practice!

Monday	Wednesday
David, Mike, Tyler	Jacob, Jonny, Katie
Andrew, Torin	Maggie, Jess
Alex, Ian, Jacob	Nate, Zach
Adam, Emily, Michael	

Recall from Feb 3: Big Picture of Course Content

1. Security of DHKE, DSA depend on DLP being hard to solve: $g^x \equiv h \mod p$

   Are there non-brute force attacks?

   - Shanks Babystep-Giantstep algorithm
     Requires storing two lists, can become impractical
   - Pollig-Hellman algorithm
     Shows why we want α to have large prime order in $\mathbb{Z}_p^*$
   - Pollard's ρ is a general collision algorithm
     More efficient in storage than Shanks in storage, but runtime may be longer

2. RSA, DHKE, DSA depend on finding large primes
   - How do you do this?
   - Can apply Pollard's ρ to factor integers

Big Picture of Course Content (cont)

3. Elliptic curve cryptography requires dramatically smaller keys than RSA, DHKE, DSA for equivalent level of security

   How does this work?

4. Security of these methods will fall apart when quantum computers are feasible

   Look at basis for lattice-based encryption schemes which have no known quantum attacks

5. Eight other topics from you!

Why is ECDLP harder than mod p DLP?

- There are algorithms faster than Shanks or Pollard's ρ for DLP that do not apply to ECDLP
  `https://en.wikipedia.org/wiki/Discrete_logarithm_records`

- This is why P-384 from exam is a safe curve for ECDHKE but a 384-bit prime for DHKE is not

Goldreich, Goldwasser, Halevi (GGH) Encryption, based on CVP

- Alice: Key creation
  - Pick good basis $\vec{v}_1, \ldots, \vec{v}_n$ and put in rows of matrix V
  - Choose matrix U with integer coefficients such that $\det(U) = \pm 1$
  - Compute bad basis as rows $\vec{w}_1, \ldots, \vec{w}_n$ of $W = UV$
  - Publish public key $\vec{w}_1, \ldots, \vec{w}_n$

Goldreich, Goldwasser, Halevi (GGH) Encryption, based on CVP

- Alice: Key creation
  - Pick good basis $\vec{v}_1, \ldots, \vec{v}_n$ and put in rows of matrix V
  - Choose matrix U with integer coefficients such that $\det(U) = \pm 1$
  - Compute bad basis as rows $\vec{w}_1, \ldots, \vec{w}_n$ of $W = UV$
  - Publish public key $\vec{w}_1, \ldots, \vec{w}_n$
- Bob: Encryption
  - Plaintext vector $\vec{m} = (m_1, \ldots, m_n) \in \mathbb{Z}^n$
  - $\vec{v} = \vec{m}W = m_1\vec{w}_1 + \cdots + m_n\vec{w}_n \in L$
  - Choose small random vector $\vec{r} \in \mathbb{R}^n$
  - Send ciphertext $\vec{e} = \vec{v} + \vec{r} \in \mathbb{R}^n$

Goldreich, Goldwasser, Halevi (GGH) Encryption, based on CVP

- Alice: Key creation
  - Pick good basis $\vec{v}_1, \ldots, \vec{v}_n$ and put in rows of matrix V
  - Choose matrix U with integer coefficients such that $\det(U) = \pm 1$
  - Compute bad basis as rows $\vec{w}_1, \ldots, \vec{w}_n$ of $W = UV$
  - Publish public key $\vec{w}_1, \ldots, \vec{w}_n$
- Bob: Encryption
  - Plaintext vector $\vec{m} = (m_1, \ldots, m_n) \in \mathbb{Z}^n$
  - $\vec{v} = \vec{m}W = m_1\vec{w}_1 + \cdots + m_n\vec{w}_n \in L$
  - Choose small random vector $\vec{r} \in \mathbb{R}^n$
  - Send ciphertext $\vec{e} = \vec{v} + \vec{r} \in \mathbb{R}^n$
- Alice: Decryption
  - Use good basis to recover $\vec{v} \in L$ *(will see details shortly)*
  - $\vec{m} = \vec{v}W^{-1}$

Example

Let $V = \begin{pmatrix} 1 & 1 & 2 \\ -3 & 1 & 1 \\ -1 & -1 & 3 \end{pmatrix}$ and $W = \begin{pmatrix} 723 & -285 & -403 \\ -691 & 273 & 385 \\ -43 & 17 & 24 \end{pmatrix}$

- Verify V and W are bases for the same lattice: $W.V^{-1} =$
- Hadamard(V) =
  Hadamard(W) =
- Bob encrypts $m = \{4, 1, 5\}$ using ephemeral $r\,\{-1, 0, 1\}$
  $v = m.W =$
  $e = v + r =$

Theorem 7.34 (Babai's Closest Vertex Algorithm)

Let $L \subset \mathbb{R}^n$ be a lattice of dimension n with basis $\mathcal{B} = \left\{\vec{v_1}, \ldots, \vec{v_n}\right\}$ and let $\vec{e} \in \mathbb{R}^n$ be an arbitrary vector.

If the basis vectors are sufficiently orthogonal, the following $\vec{v}$ solves the CVP:

- Write $\vec{e} = t_1\vec{v_1} + \cdots + t_n\vec{v_n}$ with $t_1, \ldots, t_n \in \mathbb{R}$ $\qquad (\vec{t} = \vec{e}.V^{-1})$
- Set $a_i = \lfloor t_i \rceil$ for $1 \leq i \leq n$ (i.e. round t_i) $\qquad (\vec{a} =$Round$(\vec{t})$)
- Then $\vec{v} = a_1\vec{v_1} + \cdots a_n\vec{v_n}$ $\qquad (\vec{v} = \vec{a}.V)$

Alice decrypts $e = \{1985, -782, -1106\}$ using Babai's

- $t = e.V^{-1} =$
- $a = Round(t) =$
- Closest vector: $v = a.V =$
- Recover plaintext: $m = v.W^{-1} =$

Notes on Security of GGH

- Originally suggested $L \subset \mathbb{R}^n,\ n > 300$ would be secure
- One attack is LLL-lattice reduction algorithm
  - Takes skewed public key basis and generates a more orthogonal basis for L
  - If generated basis is orthogonal enough, may be able to solve CVP
  - LLL is in the spirit of the Gram-Schmidt process, which you may have seen in Linear Algebra
  - G-S guarantees orthogonal vectors, but may result in non-integer entries
  - Mathematica command LatticeReduce[ ] implements LLL

Notes on Security of GGH (cont)

- Need to be careful with generating ephemeral $\vec{r}$
  - If send same plaintext twice with different $\vec{r}$, then gives information to break
  - If $\vec{r}$ deterministic based on $\vec{m}$, then also gives information
- There's a lot of subtlety with random number generators