Classical Crypto, Modern Crypto, and Why Number Theory is Important

June 12, 2020

Two Classic Ciphers: Vigenére and Matrix

June 12, 2020

The Vigenère cipher

Key: A word or phrase. Example: *dog* = *(3,14,6)*.
Easy to remember and transmit.
Example using *dog*.
Shift 1st letter by 3
Shift 2nd letter by 14
Shift 3nd letter by 6
Shift 4th letter by 3
Shift 5th letter by 14
Shift 6th letter by 6, etc.

Jacob Prinz is a Physics Major
Jacob Prinz isaPh ysics Major

encrypts to

MOIRP VUWTC WYDDN BGOFG SDXUU

The Vigenère cipher

Key: $k = (k_1, k_2, \ldots, k_n)$.
Encrypt (all arithmetic is mod 26)

$$Enc(m_1, m_2, \ldots, m_N) =$$

$$m_1 + k_1, m_2 + k_2, \ldots, m_n + k_n,$$

$$m_{n+1} + k_1, m_{n+2} + k_2, \ldots, m_{n+n} + k_n,$$

$$\ldots$$

Decrypt Decryption just reverse the process

The Vigenère cipher

- Size of key space?
  - If keys are 14-char then key space size $26^{14} \approx 2^{66}$
  - If variable length keys, even more.
  - Brute-force search infeasible
- Is the Vigenère cipher secure?
- Believed secure for many years...
- Might not have even been secure then...
- Easily cracked by 1900. Prob much earlier.

The Matrix Cipher

Definition: Matrix Cipher. Pick M a 2×2 matrix.

1. Encrypt via $xy \to M(xy)$.
2. Decrypt via $xy \to M^{-1}(xy)$

Encode: Break T into blocks of 2, apply M to each pair.

Decode: Do the same only with M^{-1}. Need M^{-1} to exist. It does if det is rel prime to 26.

The Matrix Cipher

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Good News:

1. Can test if M^{-1} exists, and is so find it, easily.
2. M small, so Key small.
3. Applying M or M^{-1} to a vector is easy computationally.

Bad News:

1. Eve CAN crack using frequencies of pairs of letters.
2. Eve CAN crack – Key space has $< 26^4 = 456976$. Small.

So what to do?
Use bigger matrix!

Is Matrix Cipher Uncrackable?

Use $n \times n$ matrix for large n. Say 50. Still quite feasible for Alice and Bob.

VOTE: Yes, No, Unknown to Science, Other.

Is Matrix Cipher Uncrackable?

Use $n \times n$ matrix for large n. Say 50. Still quite feasible for Alice and Bob.

VOTE: Yes, No, Unknown to Science, Other.

1. If Eve just has ciphertext then brute force needs of 26^{n^2} possibilities. Can get that down to 26^n.

Is Matrix Cipher Uncrackable?

Use $n \times n$ matrix for large n. Say 50. Still quite feasible for Alice and Bob.

VOTE: Yes, No, Unknown to Science, Other.

1. If Eve just has ciphertext then brute force needs of 26^{n^2} possibilities. Can get that down to 26^n.
2. 26^n is still large. Can Eve do better?

Is Matrix Cipher Uncrackable?

Use $n \times n$ matrix for large n. Say 50. Still quite feasible for Alice and Bob.

VOTE: Yes, No, Unknown to Science, Other.

1. If Eve just has ciphertext then brute force needs of 26^{n^2} possibilities. Can get that down to 26^n.
2. 26^n is still large. Can Eve do better?
   Seems to be **Unknown to Science!**

Is Matrix Cipher Uncrackable?

Use $n \times n$ matrix for large n. Say 50. Still quite feasible for Alice and Bob.

VOTE: Yes, No, Unknown to Science, Other.

1. If Eve just has ciphertext then brute force needs of 26^{n^2} possibilities. Can get that down to 26^n.
2. 26^n is still large. Can Eve do better?
   Seems to be **Unknown to Science!**
   So why is it not used? Discuss!

Is Matrix Cipher Uncrackable?

Use $n \times n$ matrix for large n. Say 50. Still quite feasible for Alice and Bob.

VOTE: Yes, No, Unknown to Science, Other.

1. If Eve just has ciphertext then brute force needs of 26^{n^2} possibilities. Can get that down to 26^n.
2. 26^n is still large. Can Eve do better?
   Seems to be **Unknown to Science!**
   So why is it not used? Discuss!
3. In reality Eve has prior messages and what they coded to, so from that she can easily crack it. (Next Slide.) **That is why not used.**

Cracking Matrix Cipher

Example using 2×2 Matrix Cipher.
Eve learns that (19,8) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 19 \\ 8 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

So

$$\begin{aligned} 19a + 8b &= 3 \\ 19c + 8d &= 9 \end{aligned}$$

Two linear equations, Four variables

Cracking Matrix Cipher

Example using 2×2 Matrix Cipher.
Eve learns that (19,8) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 19 \\ 8 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

So

$$\begin{aligned} 19a + 8b &= 3 \\ 19c + 8d &= 9 \end{aligned}$$

Two linear equations, Four variables

If Eve learns one more 2-letter message decoding then she will have

Four linear equations, Four variables

which she can solve! Yeah? Boo? Depends whose side you are on.

Public Key Crypto: Math Needed and DH

Private-Key Ciphers

What do the following **Private Key Encryption Schemes** all have in common:

1. Shift Cipher
2. Affine Cipher
3. Vig Cipher
4. General Sub
5. Matrix Cipher
6. One-Time Pad (this is uncrackable! but hard to use).

Alice and Bob need to **meet!** (Hence **Private Key**.)
Can Alice and Bob to establish a key without meeting?

Private-Key Ciphers

What do the following **Private Key Encryption Schemes** all have in common:

1. Shift Cipher
2. Affine Cipher
3. Vig Cipher
4. General Sub
5. Matrix Cipher
6. One-Time Pad (this is uncrackable! but hard to use).

Alice and Bob need to **meet!** (Hence **Private Key**.)
Can Alice and Bob to establish a key without meeting?
Yes! And that is the key to public-key cryptography.

Private-Key Ciphers

What do the following **Private Key Encryption Schemes** all have in common:

1. Shift Cipher
2. Affine Cipher
3. Vig Cipher
4. General Sub
5. Matrix Cipher
6. One-Time Pad (this is uncrackable! but hard to use).

Alice and Bob need to **meet!** (Hence **Private Key**.)
Can Alice and Bob to establish a key without meeting?
Yes! And that is the key to public-key cryptography.
And Public Key Crypto is the Key to Modern Cryptography.

Math Needed for Diffie-Helman

Notation

Let p be a prime.

1. $\mathbb{Z}_p$ is the numbers $\{0, \ldots, p-1\}$ with modular addition and multiplication.
2. $\mathbb{Z}_p^*$ is the numbers $\{1, \ldots, p-1\}$ with modular multiplication.

Exponentiation mod p

Example of a Good Algorithm

Want $3^{64} \pmod{101}$. All arithmetic is mod 101.

$x_0 = 3$

$x_1 = x_0^2 \equiv 9$ This is 3^2.

$x_2 = x_1^2 \equiv 9^2 \equiv 81$. This is 3^4.

$x_3 = x_2^2 \equiv 81^2 \equiv 97$. This is 3^8.

$x_4 = x_3^2 \equiv 97^2 \equiv 16$. This is 3^{16}.

$x_5 = x_4^2 \equiv 16^2 \equiv 54$. This is 3^{32}.

$x_6 = x_5^2 \equiv 54^2 \equiv 88$. This is 3^{64}.

So in 6 steps we got the answer!

Exponentiation mod p

Example of a Good Algorithm

Want $3^{64} \pmod{101}$. All arithmetic is mod 101.

$x_0 = 3$
$x_1 = x_0^2 \equiv 9$ This is 3^2.
$x_2 = x_1^2 \equiv 9^2 \equiv 81$. This is 3^4.
$x_3 = x_2^2 \equiv 81^2 \equiv 97$. This is 3^8.
$x_4 = x_3^2 \equiv 97^2 \equiv 16$. This is 3^{16}.
$x_5 = x_4^2 \equiv 16^2 \equiv 54$. This is 3^{32}.
$x_6 = x_5^2 \equiv 54^2 \equiv 88$. This is 3^{64}.
So in 6 steps we got the answer!

Generalize Repeated squaring Alg for $a^n \pmod{p}$, even if n is not a power of 2.

Exponentiation mod p

Example of a Good Algorithm

Want $3^{64} \pmod{101}$. All arithmetic is mod 101.

$x_0 = 3$

$x_1 = x_0^2 \equiv 9$ This is 3^2.

$x_2 = x_1^2 \equiv 9^2 \equiv 81$. This is 3^4.

$x_3 = x_2^2 \equiv 81^2 \equiv 97$. This is 3^8.

$x_4 = x_3^2 \equiv 97^2 \equiv 16$. This is 3^{16}.

$x_5 = x_4^2 \equiv 16^2 \equiv 54$. This is 3^{32}.

$x_6 = x_5^2 \equiv 54^2 \equiv 88$. This is 3^{64}.

So in 6 steps we got the answer!

Generalize Repeated squaring Alg for $a^n \pmod{p}$, even if n is not a power of 2.

How many steps?

Exponentiation mod p

Example of a Good Algorithm

Want $3^{64} \pmod{101}$. All arithmetic is mod 101.

$x_0 = 3$
$x_1 = x_0^2 \equiv 9$ This is 3^2.
$x_2 = x_1^2 \equiv 9^2 \equiv 81$. This is 3^4.
$x_3 = x_2^2 \equiv 81^2 \equiv 97$. This is 3^8.
$x_4 = x_3^2 \equiv 97^2 \equiv 16$. This is 3^{16}.
$x_5 = x_4^2 \equiv 16^2 \equiv 54$. This is 3^{32}.
$x_6 = x_5^2 \equiv 54^2 \equiv 88$. This is 3^{64}.
So in 6 steps we got the answer!

Generalize Repeated squaring Alg for $a^n \pmod{p}$, even if n is not a power of 2.

How many steps? $\lg n$. Fast!

Diffie-Helman Key Exchange

Generators mod p

Lets take powers of 3 mod 7. All arithmetic is mod 7.

$3^0 \equiv 1$

$3^1 \equiv 3$

$3^2 \equiv 3 \times 3^1 \equiv 9 \equiv 2$

$3^3 \equiv 3 \times 3^2 \equiv 3 \times 2 \equiv 6$

$3^4 \equiv 3 \times 3^3 \equiv 3 \times 6 \equiv 18 \equiv 4$

$3^5 \equiv 3 \times 3^4 \equiv 3 \times 4 \equiv 12 \equiv 5$

$3^6 \equiv 3 \times 3^5 \equiv 3 \times 5 \equiv 15 \equiv 1$

$$\{3^0, 3^1, 3^2, 3^3, 3^4, 3^5, 3^6\} = \{1, 2, 3, 4, 5, 6\} \text{ Not in order}$$

Generators mod p

Lets take powers of 3 mod 7. All arithmetic is mod 7.
$3^0 \equiv 1$
$3^1 \equiv 3$
$3^2 \equiv 3 \times 3^1 \equiv 9 \equiv 2$
$3^3 \equiv 3 \times 3^2 \equiv 3 \times 2 \equiv 6$
$3^4 \equiv 3 \times 3^3 \equiv 3 \times 6 \equiv 18 \equiv 4$
$3^5 \equiv 3 \times 3^4 \equiv 3 \times 4 \equiv 12 \equiv 5$
$3^6 \equiv 3 \times 3^5 \equiv 3 \times 5 \equiv 15 \equiv 1$

$$\{3^0, 3^1, 3^2, 3^3, 3^4, 3^5, 3^6\} = \{1, 2, 3, 4, 5, 6\} \text{ Not in order}$$

3 is a **generator** for $\mathbb{Z}_7$.

Generators mod p

Lets take powers of 3 mod 7. All arithmetic is mod 7.
$3^0 \equiv 1$
$3^1 \equiv 3$
$3^2 \equiv 3 \times 3^1 \equiv 9 \equiv 2$
$3^3 \equiv 3 \times 3^2 \equiv 3 \times 2 \equiv 6$
$3^4 \equiv 3 \times 3^3 \equiv 3 \times 6 \equiv 18 \equiv 4$
$3^5 \equiv 3 \times 3^4 \equiv 3 \times 4 \equiv 12 \equiv 5$
$3^6 \equiv 3 \times 3^5 \equiv 3 \times 5 \equiv 15 \equiv 1$

$$\{3^0, 3^1, 3^2, 3^3, 3^4, 3^5, 3^6\} = \{1, 2, 3, 4, 5, 6\} \text{ Not in order}$$

3 is a **generator** for $\mathbb{Z}_7$.
Definition: If p is a prime and
$\{g^0, g^1, \ldots, g^{p-1}\} = \{1, \ldots, p-1\}$ then g is a **generator** for $\mathbb{Z}_p$.

Discrete Log-Example

Fact: 5 is a generator mod 73. All arithmetic is mod 73.

Find x **such that** $5^x \equiv 26$

Discrete Log-Example

Fact: 5 is a generator mod 73. All arithmetic is mod 73.

Find x such that $5^x \equiv 26$

I do not know the answer!

Discrete Log-Example

Fact: 5 is a generator mod 73. All arithmetic is mod 73.

Find x such that $5^x \equiv 26$

I do not know the answer!

Coud try computing $5^3, 5^4, \ldots,$ until you get 26.
Might take ~ 70 steps.

Discrete Log-General

Definition Let p be a prime and g be a generator mod p.
The **Discrete Log Problem** is:
given y, find x such that $g^x = y$.

1. If $g, y \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ then problem suspected hard.
2. Obv alg: $O(p)$ steps. There is an $O(\sqrt{p})$ alg. Still too slow.

Consider What We Already Have Here

- Exponentiation is Easy.
- Discrete Log is thought to be Hard.

Consider What We Already Have Here

- Exponentiation is Easy.
- Discrete Log is thought to be Hard.

Can we come up with a crypto system where Alice and Bob do Exponentiation to encrypt and decrypt, while Eve has to do Discrete Log to crack it?

Consider What We Already Have Here

- Exponentiation is Easy.
- Discrete Log is thought to be Hard.

Can we come up with a crypto system where Alice and Bob do Exponentiation to encrypt and decrypt, while Eve has to do Discrete Log to crack it?

No. But we'll come close.

Other Things Needed

Need Alice and Bob to be able to

1. Find Large Primes
2. Find generators for those primes.

Both are fast if done together:
Find p such that p prime AND $\frac{p-1}{2}$ is prime.
Then finding generator is easy.

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s. n is sec param.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice broadcasts $(p, g, HAHA)$.
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and broadcasts $(g^a, HAHA)$.
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and broadcasts $(g^b, HAHA)$.
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s. n is sec param.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice broadcasts $(p, g, HAHA)$.
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and broadcasts $(g^a, HAHA)$.
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and broadcasts $(g^b, HAHA)$.
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

PRO: Alice and Bob can execute the protocol easily.

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s. n is sec param.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice broadcasts $(p, g, HAHA)$.
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and broadcasts $(g^a, HAHA)$.
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and broadcasts $(g^b, HAHA)$.
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

PRO: Alice and Bob can execute the protocol easily.
Biggest PRO: Alice and Bob never had to meet!

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s. n is sec param.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice broadcasts $(p, g, HAHA)$.
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and broadcasts $(g^a, HAHA)$.
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and broadcasts $(g^b, HAHA)$.
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

PRO: Alice and Bob can execute the protocol easily.
Biggest PRO: Alice and Bob never had to meet!
Question: Can Eve find out s?

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s. n is sec param.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice broadcasts $(p, g, HAHA)$.
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and broadcasts $(g^a, HAHA)$.
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and broadcasts $(g^b, HAHA)$.
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

PRO: Alice and Bob can execute the protocol easily.
Biggest PRO: Alice and Bob never had to meet!
Question: Can Eve find out s?
If Eve can compute Discrete Log problem then Yes.

Converse is not known.

DH and RSA

DH is a cryptosystem that Alice and Bob can use to generate a shared secret key.

DH and RSA

DH is a cryptosystem that Alice and Bob can use to generate a shared secret key.

DH cannot be used to SEND messages. It can be used to establish a key so that Alice and Bob CAN send messages.

DH and RSA

DH is a cryptosystem that Alice and Bob can use to generate a shared secret key.

DH cannot be used to SEND messages. It can be used to establish a key so that Alice and Bob CAN send messages.

RSA is a cryptosystem that Alice and Bob can use to send messages.

RSA

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.
2. If Factoring is easy then RSA can be cracked.

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.
2. If Factoring is easy then RSA can be cracked.
3. Converse is not known.

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.
2. If Factoring is easy then RSA can be cracked.
3. Converse is not known.
4. We will not be discussing RSA past this slide.

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.
2. If Factoring is easy then RSA can be cracked.
3. Converse is not known.
4. We will not be discussing RSA past this slide.
5. Shamir is an Israeli citizen and recently could not get a Visa to go to the RSA conference.

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.
2. If Factoring is easy then RSA can be cracked.
3. Converse is not known.
4. We will not be discussing RSA past this slide.
5. Shamir is an Israeli citizen and recently could not get a Visa to go to the RSA conference.
6. Would be funny if it wasn't sad.

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.
2. If Factoring is easy then RSA can be cracked.
3. Converse is not known.
4. We will not be discussing RSA past this slide.
5. Shamir is an Israeli citizen and recently could not get a Visa to go to the RSA conference.
6. Would be funny if it wasn't sad.
7. My friends in the real world tell me that RSA is used much more than Diffie Helman.

RSA

1. Rivest-Shamir-Adelman in 1978 came up with RSA, another crypto system.
2. If Factoring is easy then RSA can be cracked.
3. Converse is not known.
4. We will not be discussing RSA past this slide.
5. Shamir is an Israeli citizen and recently could not get a Visa to go to the RSA conference.
6. Would be funny if it wasn't sad.
7. My friends in the real world tell me that RSA is used much more than Diffie Helman.
8. I have no friends in the real world, so statement is true vacuously.

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

1. Clever number theory has lead to algorithms for Factoring with run times around $2^{L^{1/3}\log(L)^{2/3}}$ where L is the Length of the number.

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

1. Clever number theory has lead to algorithms for Factoring with run times around $2^{L^{1/3}\log(L)^{2/3}}$ where L is the Length of the number.
2. Some of these have been coded up and some work well on parallel machines.

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

1. Clever number theory has lead to algorithms for Factoring with run times around $2^{L^{1/3}\log(L)^{2/3}}$ where L is the Length of the number.
2. Some of these have been coded up and some work well on parallel machines.
3. NO. The NSA does not have brilliant Number theorists who have secret algorithms for factoring in P.

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

1. Clever number theory has lead to algorithms for Factoring with run times around $2^{L^{1/3}\log(L)^{2/3}}$ where L is the Length of the number.
2. Some of these have been coded up and some work well on parallel machines.
3. NO. The NSA does not have brilliant Number theorists who have secret algorithms for factoring in P. They just use Facebook to track bad people.

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

1. Clever number theory has lead to algorithms for Factoring with run times around $2^{L^{1/3}\log(L)^{2/3}}$ where L is the Length of the number.
2. Some of these have been coded up and some work well on parallel machines.
3. NO. The NSA does not have brilliant Number theorists who have secret algorithms for factoring in P. They just use Facebook to track bad people. And good people.

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

1. Clever number theory has lead to algorithms for Factoring with run times around $2^{L^{1/3} \log(L)^{2/3}}$ where L is the Length of the number.
2. Some of these have been coded up and some work well on parallel machines.
3. NO. The NSA does not have brilliant Number theorists who have secret algorithms for factoring in P. They just use Facebook to track bad people. And good people.
4. There are fast quantum algorithms. So far these are theoretical only.

Where Are We Now?

1. If Discrete Log is hard then Diffie Helman is uncrackable.
2. If Factoring is hard then RSA is uncrackable.

Sounds good but:

1. Clever number theory has lead to algorithms for Factoring with run times around $2^{L^{1/3}\log(L)^{2/3}}$ where L is the Length of the number.
2. Some of these have been coded up and some work well on parallel machines.
3. NO. The NSA does not have brilliant Number theorists who have secret algorithms for factoring in P. They just use Facebook to track bad people. And good people.
4. There are fast quantum algorithms. So far these are theoretical only.
5. Hence the need for crypto systems based on OTHER assumptions.

One More Application that Needs Discrete Log Hard

Yao's Millionaire's Problem

1. Donald has x dollars
2. Warren has y dollars.
3. They want to know who has more money.
4. They don't want to reveal their worth to the other.
5. Yao came up with a protocol that will reveal to both who has more money but will not reveal to either how much the other has ASSUMING that neither one can do Discrete Log fast.

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.

Several are based on problems with Lattices being hard.

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.
Several are based on problems with Lattices being hard.
Hence the name Lattice-Theoretic Crytography.

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.
Several are based on problems with Lattices being hard.
Hence the name Lattice-Theoretic Crytography.
Project:

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.
Several are based on problems with Lattices being hard.
Hence the name Lattice-Theoretic Crytography.
Project:

1. Learn cryptosystems based on lattice problems being hard.

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.
Several are based on problems with Lattices being hard.
Hence the name Lattice-Theoretic Crytography.
Project:

1. Learn cryptosystems based on lattice problems being hard.
2. Code them up.

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.
Several are based on problems with Lattices being hard.
Hence the name Lattice-Theoretic Crytography.
Project:

1. Learn cryptosystems based on lattice problems being hard.
2. Code them up.
3. See how they do for time, for security, for ease of use, for ease of coding.

REU Project

There are Cryptosystems that are NOT based on Number Theoretic Problems being hard.
Several are based on problems with Lattices being hard.
Hence the name Lattice-Theoretic Crytography.
Project:

1. Learn cryptosystems based on lattice problems being hard.
2. Code them up.
3. See how they do for time, for security, for ease of use, for ease of coding.
4. Use to keep America safe!