Secret Sharing

A **secret** is an n-bit string.
Throughout this talk assume that Zelda has a secret $s \in \{0,1\}^n$.
She will want to give **shares of the secret** to various people.

Applications

Rumor: Secret Sharing is used for the Russian Nuclear Codes. There are three people (one is Putin) and if two of them agree to launch, they can launch.

Fact: If t out of n people need to sign a document then secret sharing is used as a building block in the protocol.

Secret Sharing: Four People, Need Three

A_1, A_2, A_3, A_4 such that

1. If all four of A_1, A_2, A_3, A_4 get together they can find s.
2. If any three of them get together then learn **NOTHING**.

One Scheme

1. Zelda breaks s up into $s = s_1 s_1 s_3 s_4$ where

$$|s_1| = |s_2| = |s_3| = |s_4| = \frac{n}{4}$$

2. Zelda gives A_i the string s_i.

Does this work?

One Scheme

1. Zelda breaks s up into $s = s_1 s_1 s_3 s_4$ where

$$|s_1| = |s_2| = |s_3| = |s_4| = \frac{n}{4}$$

2. Zelda gives A_i the string s_i.

Does this work?

1. If A_1, A_2, A_3, A_4 get together they can find s. **YES!!**

One Scheme

1. Zelda breaks s up into $s = s_1 s_1 s_3 s_4$ where

$$|s_1| = |s_2| = |s_3| = |s_4| = \frac{n}{4}$$

2. Zelda gives A_i the string s_i.

Does this work?

1. If A_1, A_2, A_3, A_4 get together they can find s. **YES!!**
2. If any three of them get together they learn **NOTHING**. **NO**.
   - 2.1 A_1 learns s_1 which is $\frac{1}{4}$ **of the secret!**
   - 2.2 A_1, A_2 learn $s_1 s_2$ which is $\frac{1}{2}$ **of the secret!**
   - 2.3 A_1, A_2, A_3 learn $s_1 s_2 s_3$ which is $\frac{3}{4}$ **of the secret!**

What do we mean by **NOTHING**?

If any three of them get together they learn **NOTHING**
Informally:

1. Before Zelda gives out shares, if any three A_i, A_j, A_k get together, they know $BLAH_{i,j,k}$.
2. After Zelda gives out shares, if any three A_i, A_j, A_k get together, they know $BLAH_{i,j,k}$.
3. Giving out the shares tells each triple **NOTHING** they did not already know.

If A_i, A_j, A_k have **unlimited computing power**

What do we mean by **NOTHING**?

If any three of them get together they learn **NOTHING**
Informally:

1. Before Zelda gives out shares, if any three A_i, A_j, A_k get together, they know $BLAH_{i,j,k}$.
2. After Zelda gives out shares, if any three A_i, A_j, A_k get together, they know $BLAH_{i,j,k}$.
3. Giving out the shares tells each triple **NOTHING** they did not already know.

If A_i, A_j, A_k have **unlimited computing power**
they still learn **NOTHING**.

What do we mean by **NOTHING**?

If any three of them get together they learn **NOTHING**
Informally:

1. Before Zelda gives out shares, if any three A_i, A_j, A_k get together, they know $BLAH_{i,j,k}$.
2. After Zelda gives out shares, if any three A_i, A_j, A_k get together, they know $BLAH_{i,j,k}$.
3. Giving out the shares tells each triple **NOTHING** they did not already know.

If A_i, A_j, A_k have **unlimited computing power**
they still learn **NOTHING**.

Information-Theoretic Security

Can Zelda Do This?

VOTE: Can Zelda Do this:

1. YES
2. NO
3. YES given some hardness assumption
4. UNKNOWN TO SCIENCE

Can Zelda Do This?

VOTE: Can Zelda Do this:

1. YES
2. NO
3. YES given some hardness assumption
4. UNKNOWN TO SCIENCE

YES

Random String Approach

Zelda gives out shares of the secret

1. Zelda has secret $s \in \{0,1\}^n$.
2. Zelda generates **random** $r_1, r_2, r_3 \in \{0,1\}^n$.
3. For $i = 1, 2, 3$ Zelda gives A_i the string $s_i = r_i$.
4. Zelda gives A_4 the string $s_4 = s \oplus r_1 \oplus r_2 \oplus r_3$

A_1, A_2, A_3 A_4 Can Recover the Secret

$$s_1 \oplus s_2 \oplus s_3 \oplus s_4 = r_1 \oplus r_2 \oplus r_3 \oplus r_1 \oplus r_2 \oplus r_3 \oplus s = s$$

Easy to see that if a triple get together they learn NOTHING
(Jon Katz says this requires a careful proof.)

Less People Needed To Recover Secret

Zelda wants to give strings to A_1, A_2, A_3, A_4 such that

1. Any TWO of A_1, A_2, A_3, A_4 can find s.
2. Any ONE learns **NOTHING**.

Random String Approach

For each $1 \leq i < j \leq 4$

1. Zelda generates **random** $r \in \{0,1\}^n$.
2. Zelda gives A_i the strings $s_{i,(i,j)} = (i, j, r)$.
3. Zelda gives A_j the strings $s_{j,(i,j)} = (i, j, r \oplus s)$.

A_i, A_j Can Recover the Secret

A_i takes (i, j, r) and just uses the r.

A_j takes $(i, j, r \oplus s)$ and just uses the $r \oplus s$.

They both compute $r \oplus r \oplus s = s$.

Easy to see that one person learns NOTHING

(Jon Katz says this needs a rigorous proof.)

How Many Strings Does A_i Get?

A_1 gets strings $(1, 2, r)$, $(1, 3, r)$. So two strings.
A_i gets one string for every group she is a part of.

Zelda wants to give strings to $A_1, \ldots, A_{10}$ such that

1. Any FIVE of $A_1, \ldots, A_{10}$ can find s.
2. Any FOUR learn **NOTHING**.

How many strings does A_1 get?
$(1, a_1, a_2, a_3, a_4)$ where $2 \leq a_1 < a_2 < a_3 < a_4 \leq 10$.
$\binom{9}{4} = 126$ strings.

How Many Strings Does A_i Get?

Zelda wants to give strings to $A_1, \ldots, A_L$ such that

1. Any $L/2$ of $A_1, \ldots, A_L$ can find s.
2. Any $L/2 - 1$ learn **NOTHING**.

How many strings does A_1 get?
$(1, a_1, \ldots, a_{L/2-1})$ where $2 \leq a_1 < \cdots < a_{L/2-1} \leq L$.
$\binom{L}{L/2} \sim \frac{2^L}{\sqrt{L}}$ strings. **Thats a lot of strings!**
VOTE

1. Requires roughly 2^L strings.
2. $O(\alpha^L)$ strings for some $1 < \alpha < 2$ but not poly.
3. $O(L^a)$ strings for some $a > 1$ but not linear.
4. $O(1)$ strings but the number of strings can be very large.
5. $O(1)$ strings but the number of strings is always < 10.

How Many Strings Does A_i Get?

Zelda wants to give strings to $A_1, \ldots, A_L$ such that

1. Any $L/2$ of $A_1, \ldots, A_L$ can find s.
2. Any $L/2 - 1$ learn **NOTHING**.

How many strings does A_1 get?
$(1, a_1, \ldots, a_{L/2-1})$ where $2 \leq a_1 < \cdots < a_{L/2-1} \leq L$.
$\binom{L}{L/2} \sim \frac{2^L}{\sqrt{L}}$ strings. **Thats a lot of strings!**
VOTE

1. Requires roughly 2^L strings.
2. $O(\alpha^L)$ strings for some $1 < \alpha < 2$ but not poly.
3. $O(L^a)$ strings for some $a > 1$ but not linear.
4. $O(1)$ strings but the number of strings can be very large.
5. $O(1)$ strings but the number of strings is always < 10.

You can always do this problem with 1 string. Really!

Secret Sharing With Polynomials

Zelda wants to give strings to $A_1, A_2, A_3, A_4, A_5, A_6$ such that

Any 3 of $A_1, A_2, A_3, A_4, A_5, A_6$ can find s.
Any 2 learn **NOTHING**.

1. Secret s. Zelda picks prime $p \sim s$, Zelda works mod p.
2. Zelda generates RANDOM numbers $a_2, a_1 \in \{0, \ldots, p-1\}$
3. Zelda forms polynomial $q(x) = a_2x^2 + a_1x + s$.
4. Zelda gives A_1 $q(1)$, A_2 $q(2)$, ..., A_6 $q(6)$ (all mod p). These are all of length $\sim |s|$.

1. Any 3 have 3 points from $q(x)$ so can find $q(x)$, s.
2. Any 2 have 2 points from $q(x)$. Constant term (s) **anything!**.

Example

$s = 20$. We'll use $p = 23$.

1. Zelda picks $a_2 = 8$ and $a_1 = 13$.
2. Zelda forms polynomial $q(x) = 8x^2 + 13x + 20$.
3. Zelda gives A_1 $q(1) = 18$, A_2 $q(2) = 9$, A_3 $q(3) = 16$, A_4 $q(4) = 16$, A_5 $q(5) = 9$, A_6 $q(6) = 18$.

If A_1, A_3, A_4 get together and want to find $q(x)$ hence s.
$q(x) = a_2x^2 + a_1x + s$.
$q(1) = 18$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 18 \pmod{23}$
$q(3) = 16$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 16 \pmod{23}$
$q(4) = 16$: $a_2 \times 4^2 + a_1 \times 4 + s \equiv 16 \pmod{23}$
3 linear equations in, 3 variable, over mod 23 can be solved.

A Note About Linear Equations

We claim:

$a_2 \times 1^2 + a_1 \times 1 + s \equiv 18 \pmod{23}$

$a_2 \times 3^2 + a_1 \times 3 + s \equiv 16 \pmod{23}$

$a_2 \times 4^2 + a_1 \times 4 + s \equiv 16 \pmod{23}$

3 linear equations in, 3 variable, over mod 23 can be solved.

Could we have solved this had we used mod 24?

VOTE

1. YES
2. NO

A Note About Linear Equations

We claim:
$a_2 \times 1^2 + a_1 \times 1 + s \equiv 18 \pmod{23}$
$a_2 \times 3^2 + a_1 \times 3 + s \equiv 16 \pmod{23}$
$a_2 \times 4^2 + a_1 \times 4 + s \equiv 16 \pmod{23}$
3 linear equations in, 3 variable, over mod 23 can be solved.

Could we have solved this had we used mod 24?
VOTE

1. YES
2. NO

NO
Need a domain where every number has a mult inverse.
Over mod p, p primes, all numbers have mult inverses.
Mod 24 no even number has an mult inverse.

Threshold Secret Sharing With Polynomials

Zelda wants to give strings to $A_1, \ldots, A_L$ such that

Any t of $A_1, \ldots, A_L$ can find s.
Any $t-1$ learn **NOTHING**.

1. Secret s. Zelda picks prime $p \sim s$, Zelda works mod p.
2. Zelda generates RANDOM numbers $a_{t-1}, \ldots, a_1 \in \{0, \ldots, p-1\}$
3. Zelda forms polynomial $q(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + s$.
4. For $1 \leq i \leq L$ Zelda gives A_i $q(i)$ mod p.

1. Any t have t points of $q(x)$ so can find $q(x)$ and s.
2. Any $t-1$ have $t-1$ points of $q(x)$. Constant term (s) could be **anything!**.

Caveats

Known: For all numbers s there exists a prime p with $p \leq 2s$.

Secret is s. Let $|s| = n$,the LENGTH of s. $|2s| = n + 1$.

Upshot: The secret is length n, the shares are of length $n + 1$.

Good News: Every A_i gets ONE share.

Bad News: That share is of length $n + 1$, not n.

VOTE: Can Zelda do threshold secret sharing where every student gets ONE share of length n?

1. YES
2. NO
3. YES given some hardness assumption
4. UNKNOWN TO SCIENCE

Caveats

Known: For all numbers s there exists a prime p with $p \leq 2s$.

Secret is s. Let $|s| = n$,the LENGTH of s. $|2s| = n + 1$.

Upshot: The secret is length n, the shares are of length $n + 1$.

Good News: Every A_i gets ONE share.

Bad News: That share is of length $n + 1$, not n.

VOTE: Can Zelda do threshold secret sharing where every student gets ONE share of length n?

1. YES
2. NO
3. YES given some hardness assumption
4. UNKNOWN TO SCIENCE

YES

Why Did We Use Primes?

We used $\{0, 1, \ldots, p-1\}$ since to interpolate a polynomial we need to be able to take inverses.

Definition: A **Field** is a set F together with operations $+, \times$ such that

1. There is a 0 element such that $(\forall x)[x + 0 = x]$.
2. There is a 1 element such that $(\forall x)[x \times 1 = x]$.
3. $(\forall x, y)[x + y = y + x \wedge x \times y = y \times x]$.
4. $(\forall x, y, z)[x \times (y + z) = x \times y + x \times z]$.
5. $(\forall x)(\exists y)[x + y = 0]$.
6. $(\forall x \neq 0)(\exists y)[x \times y = 1]$. (This one is KEY.)

KEY: Operating over a field is EXACTLY like operations over **Q**. Hence you can interpolate polynomials.

WE USED: $\{0, \ldots, p-1\}$ with operations mod p is a field.

Can we use a different field?

KEY: There is a field of size p^a for all primes p and $a \geq 1$.

WE USE: For all n there is a field on 2^n elements.
If secret is s of length n, use the field on 2^n elements. All elements of it are of length n.

Upshot: For threshold there is a secret sharing scheme where everyone gets ONE share of size EXACTLY the size of the secret.

Can we use even shorter shares?

$|s| = n$, L people, threshold t.
Is there a Secret Sharing Scheme where someone gets share of size $< n$? We will allow others to get long shares, say, $\Omega(n^2 \log n)$.
VOTE

1. $(\exists)$ scheme, A_1 gets size $n-1$, all else gets size $\Omega(n^2 \log n)$.
2. $(\exists)$ scheme, A_1 gets size $\lceil n/2 \rceil$, all else gets size $\Omega(n^2 \log n)$.
3. $(\exists)$ scheme, A_1 gets size $\lceil \sqrt{n} \rceil$, all else gets size $\Omega(n^2 \log n)$.
4. $(\exists)$ scheme, A_1 gets size $\lceil \log n \rceil$, all else gets size $\Omega(n^2 \log n)$.
5. NO- in ANY scheme A_1 MUST get size $\geq n$.

Can we use even shorter shares?

$|s| = n$, L people, threshold t.
Is there a Secret Sharing Scheme where someone gets share of size $< n$? We will allow others to get long shares, say, $\Omega(n^2 \log n)$.
VOTE

1. ($\exists$) scheme, A_1 gets size $n-1$, all else gets size $\Omega(n^2 \log n)$.
2. ($\exists$) scheme, A_1 gets size $\lceil n/2 \rceil$, all else gets size $\Omega(n^2 \log n)$.
3. ($\exists$) scheme, A_1 gets size $\lceil \sqrt{n} \rceil$, all else gets size $\Omega(n^2 \log n)$.
4. ($\exists$) scheme, A_1 gets size $\lceil \log n \rceil$, all else gets size $\Omega(n^2 \log n)$.
5. NO- in ANY scheme A_1 MUST get size $\geq n$.

NO

CANNOT give anyone shares $< n$

$|s| = n$, L people, threshold t.
Assume there is a scheme were A_1 gets a share of size $n-1$.
We claim that $t-1$ people can get together and learn...**SOMETHING**.

$A_2, \ldots, A_t$: They pool their shares. They know that A_1 has a share of size $n-1$. They can go through **every** possible $w \in \{0,1\}^{n-1}$ that could be a share of A_1.

For each $w \in \{0,1\}^{n-1}$ they find what s would be IF w was A_1's share.

They find 2^{n-1} possibilities for what s is.
This is SOMETHING! They have ELIMINATED half of the possibilities for the secret.

Are Shorter Shares Possible?

If we **demand** info-security then **everyone** gets a share $\geq n$.
What if we only **demand** comp-security?
VOTE

1. Can get shares $< \alpha n$ assuming Secure Symmetric-Key Encryption.
2. Even with hardness assumption REQUIRES shares $\geq n$.

Are Shorter Shares Possible?

If we **demand** info-security then **everyone** gets a share $\geq n$.
What if we only **demand** comp-security?
VOTE

1. Can get shares $< \alpha n$ assuming Secure Symmetric-Key Encryption.
2. Even with hardness assumption REQUIRES shares $\geq n$.

Can get shares $< \alpha n$ assuming Secure Symmetric-Key Encryption.

Question

We showed that Threshold Secret Sharing had shares of length n.

What else does?

Definition An **Access Structure** is a subset of $\{A_1, \ldots, A_k\}$ closed under superset. E.g.: at least t people.

Which access structures admit a scheme with shares of length n?

Access Structures that admit Scheme with Share Length n

1. Threshold Secret sharing: if t or more get together.
2. Let G be a graph. Let s, t be nodes. People are at every node. Any connected path can get the secret.
3. Monotone Boolean Formulas where each variable occurs once. Example: $(A_1 \vee A_2) \wedge (A_3 \vee A_4 \vee A_5)$ means any set with at least one from $\{A_1, A_2\}$ and at least one from $\{A_3, A_4, A_5\}$.
4. Monotone Span Programs (omitted but has to do with matrices).

Access Structures that do not admit Scheme with Share Length n

1. $(A_1 \wedge A_2) \vee (A_2 \wedge A_3) \vee (A_3 \wedge A_4)$
2. $(A_1 \wedge A_2 \wedge A_3) \vee (A_1 \wedge A_4) \vee (A_2 \wedge A_4) \vee (A_3 \vee A_4)$ (Called *captain and crew*. A_1, A_2, A_3 is the crew, and A_4 is the captain. Either the entire crew, or the captain and one crew member, can get the secret. Can extend to any number of crew members.)
3. $(A_1 \wedge A_2 \wedge A_3) \vee (A_1 \wedge A_4) \vee (A_2 \wedge A_4)$ (Called *captain and rival*. A_1, A_2, A_3 is the crew, A_3 is a rival, A_4 is the captain. Either the entire crew, or the captain and any crew member who is NOT rival, can get the secret. Can extend to any number of crew members.)
4. Any access structure that **contains** any of the above.

In all of the above all get a share of size $1.5n$ and this is optimal.

Gap Theorem

Theorem: If there is a secret sharing scheme (of a certain type) where everyone gets share of size $< 1.5n$ then there is a secret sharing scheme where everyone gets share of size n.

of a certain type? The counterexample has share size between $1.33\ldots$ and 1. It is very **funky**

Open Question

Determine for ever access structure the functions $f(n)$ and $g(n)$ such that

1. $(\exists)$ Scheme where everyone gets $\leq f(n)$ sized share.
2. $(\forall)$ Scheme someone gets $\geq g(n)$ sized share.
3. $f(n)$ and $g(n)$ are close together.