BILL START
RECORDING LECTURE

Threshold Secret Sharing: Information-Theoretic

Threshold Secret Sharing

Zelda has a **secret** $s \in \{0,1\}^n$.

Threshold Secret Sharing

Zelda has a **secret** $s \in \{0,1\}^n$.

Def Let $1 \leq t \leq m$. **(t,m)-secret sharing** is a way for Zelda to give strings to $A_1, \ldots, A_m$ such that:

Threshold Secret Sharing

Zelda has a **secret** $s \in \{0,1\}^n$.

Def Let $1 \le t \le m$. **(t,m)-secret sharing** is a way for Zelda to give strings to $A_1, \ldots, A_m$ such that:

1. If any t get together then they can learn s.

Threshold Secret Sharing

Zelda has a **secret** $s \in \{0, 1\}^n$.

Def Let $1 \leq t \leq m$. **(t, m)-secret sharing** is a way for Zelda to give strings to $A_1, \ldots, A_m$ such that:

1. If any t get together then they can learn s.
2. If any $t - 1$ get together they cannot learn s.

Threshold Secret Sharing

Zelda has a **secret** $s \in \{0,1\}^n$.

Def Let $1 \leq t \leq m$. **(t, m)-secret sharing** is a way for Zelda to give strings to $A_1, \ldots, A_m$ such that:

1. If any t get together then they can learn s.
2. If any $t-1$ get together they cannot learn s.

What do we mean by **Cannot learn the secret**?

Threshold Secret Sharing

Zelda has a **secret** $s \in \{0,1\}^n$.

Def Let $1 \le t \le m$. **(t,m)-secret sharing** is a way for Zelda to give strings to $A_1, \ldots, A_m$ such that:

1. If any t get together then they can learn s.
2. If any $t-1$ get together they cannot learn s.

What do we mean by **Cannot learn the secret**?

Info-theory-security. If $t-1$ people have big fancy supercomputers they cannot learn ANYTHING about s.

Threshold Secret Sharing

Zelda has a **secret** $s \in \{0,1\}^n$.

Def Let $1 \le t \le m$. **(t,m)-secret sharing** is a way for Zelda to give strings to $A_1, \ldots, A_m$ such that:

1. If any t get together then they can learn s.
2. If any $t-1$ get together they cannot learn s.

What do we mean by **Cannot learn the secret**?

Info-theory-security. If $t-1$ people have big fancy supercomputers they cannot learn ANYTHING about s.

Time permitting we look at comp-security where we assume a limitation on how much the players can compute.

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.

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 For people signing a contract long distance, secret sharing is used as a building block in the protocol.

$(4, 4)$-Secret Sharing

Zelda has a secret s. A_1, A_2, A_3, A_4 are people. We want:

$(4, 4)$-Secret Sharing

Zelda has a secret s. A_1, A_2, A_3, A_4 are people. We want:

1. If all four of A_1, A_2, A_3, A_4 get together, they can find s.

$(4, 4)$-Secret Sharing

Zelda has a secret s. A_1, A_2, A_3, A_4 are people. We want:

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 they learn **NOTHING**.

An Attempt at $(4,4)$-Secret Sharing

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ where

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

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ where
$$|s_1| = |s_2| = |s_3| = |s_4| = \frac{n}{4}$$
2. Zelda gives A_i the string s_i.

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ where

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

2. Zelda gives A_i the string s_i.

Does this work?

An Attempt at $(4, 4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ 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.

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ 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!!**

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ 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**.

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ 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**.

An Attempt at $(4, 4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ 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!**

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ 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!**

An Attempt at $(4,4)$-Secret Sharing

1. Zelda breaks s up into $s = s_1 s_2 s_3 s_4$ 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!**

Is $(4, 4)$-Secret Sharing Possible?

VOTE Is $(4, 4)$-Secret sharing possible?

Is $(4,4)$-Secret Sharing Possible?

VOTE Is $(4,4)$-Secret sharing possible?

1. YES and this is known.

Is $(4,4)$-Secret Sharing Possible?

VOTE Is $(4,4)$-Secret sharing possible?

1. YES and this is known.
2. NO and this is known.

Is $(4, 4)$-Secret Sharing Possible?

VOTE Is $(4, 4)$-Secret sharing possible?

1. YES and this is known.
2. NO and this is known.
3. YES given some hardness assumption, and this is known.

Is $(4, 4)$-Secret Sharing Possible?

VOTE Is $(4, 4)$-Secret sharing possible?

1. YES and this is known.
2. NO and this is known.
3. YES given some hardness assumption, and this is known.
4. UNKNOWN TO SCIENCE!

Is $(4,4)$-Secret Sharing Possible?

VOTE Is $(4,4)$-Secret sharing possible?

1. YES and this is known.
2. NO and this is known.
3. YES given some hardness assumption, and this is known.
4. UNKNOWN TO SCIENCE!

YES

Random String Approach

Zelda gives out shares of the secret

Random String Approach

Zelda gives out shares of the secret

1. Secret $s \in \{0,1\}^n$. Zelda gen **random** $r_1, r_2, r_3 \in \{0,1\}^n$.

Random String Approach

Zelda gives out shares of the secret

1. Secret $s \in \{0,1\}^n$. Zelda gen **random** $r_1, r_2, r_3 \in \{0,1\}^n$.
2. Zelda gives A_1 $s_1 = r_1$.

Random String Approach

Zelda gives out shares of the secret

1. Secret $s \in \{0,1\}^n$. Zelda gen **random** $r_1, r_2, r_3 \in \{0,1\}^n$.
2. Zelda gives A_1 $s_1 = r_1$.
   Zelda gives A_2 $s_2 = r_2$.

Random String Approach

Zelda gives out shares of the secret

1. Secret $s \in \{0,1\}^n$. Zelda gen **random** $r_1, r_2, r_3 \in \{0,1\}^n$.
2. Zelda gives A_1 $s_1 = r_1$.
   Zelda gives A_2 $s_2 = r_2$.
   Zelda gives A_3 $s_3 = r_3$.

Random String Approach

Zelda gives out shares of the secret

1. Secret $s \in \{0,1\}^n$. Zelda gen **random** $r_1, r_2, r_3 \in \{0,1\}^n$.
2. Zelda gives A_1 $s_1 = r_1$.
   Zelda gives A_2 $s_2 = r_2$.
   Zelda gives A_3 $s_3 = r_3$.
   Zelda gives A_4 $s_4 = s \oplus r_1 \oplus r_2 \oplus r_3$.

Random String Approach

Zelda gives out shares of the secret

1. Secret $s \in \{0,1\}^n$. Zelda gen **random** $r_1, r_2, r_3 \in \{0,1\}^n$.
2. Zelda gives A_1 $s_1 = r_1$.
   Zelda gives A_2 $s_2 = r_2$.
   Zelda gives A_3 $s_3 = r_3$.
   Zelda gives A_4 $s_4 = s \oplus r_1 \oplus r_2 \oplus r_3$.

A_1, A_2, A_3, A_4 Can Recover the Secret

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

Random String Approach

Zelda gives out shares of the secret

1. Secret $s \in \{0,1\}^n$. Zelda gen **random** $r_1, r_2, r_3 \in \{0,1\}^n$.
2. Zelda gives A_1 $s_1 = r_1$.
   Zelda gives A_2 $s_2 = r_2$.
   Zelda gives A_3 $s_3 = r_3$.
   Zelda gives A_4 $s_4 = s \oplus r_1 \oplus r_2 \oplus r_3$.

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 ≤ 3 get together they learn NOTHING

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$

Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.

Idea Zelda will secret share with **every pair separately**.

$(2, 4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0, 1\}^n$
Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.
Idea Zelda will secret share with **every pair separately**.
Z Gen random r_{12}. Give A_1 $(1, 2, r_{12})$ and A_2 $(1, 2, s \oplus r_{12})$.

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$
Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.
Idea Zelda will secret share with **every pair separately**.
Z Gen random r_{12}. Give A_1 $(1, 2, r_{12})$ and A_2 $(1, 2, s \oplus r_{12})$.
Z Gen random r_{13}. Give A_1 $(1, 3, r_{13})$ and A_3 $(1, 3, s \oplus r_{13})$.

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$
Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.
Idea Zelda will secret share with **every pair separately**.
Z Gen random r_{12}. Give A_1 $(1,2,r_{12})$ and A_2 $(1,2,s \oplus r_{12})$.
Z Gen random r_{13}. Give A_1 $(1,3,r_{13})$ and A_3 $(1,3,s \oplus r_{13})$.
Z Gen random r_{14}. Give A_1 $(1,4,r_{14})$ and A_4 $(1,4,s \oplus r_{14})$.

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$

Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.

Idea Zelda will secret share with **every pair separately**.

Z Gen random r_{12}. Give A_1 $(1,2,r_{12})$ and A_2 $(1,2,s \oplus r_{12})$.

Z Gen random r_{13}. Give A_1 $(1,3,r_{13})$ and A_3 $(1,3,s \oplus r_{13})$.

Z Gen random r_{14}. Give A_1 $(1,4,r_{14})$ and A_4 $(1,4,s \oplus r_{14})$.

Z Gen random r_{23}. Give A_2 $(2,3,r_{23})$ and A_3 $(2,3,s \oplus r_{23})$.

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$
Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.
Idea Zelda will secret share with **every pair separately**.
Z Gen random r_{12}. Give A_1 $(1,2,r_{12})$ and A_2 $(1,2,s \oplus r_{12})$.
Z Gen random r_{13}. Give A_1 $(1,3,r_{13})$ and A_3 $(1,3,s \oplus r_{13})$.
Z Gen random r_{14}. Give A_1 $(1,4,r_{14})$ and A_4 $(1,4,s \oplus r_{14})$.
Z Gen random r_{23}. Give A_2 $(2,3,r_{23})$ and A_3 $(2,3,s \oplus r_{23})$.
Z Gen random r_{24}. Give A_2 $(2,4,r_{24})$ and A_4 $(2,4,s \oplus r_{24})$.

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$
Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.
Idea Zelda will secret share with **every pair separately**.
Z Gen random r_{12}. Give A_1 $(1,2,r_{12})$ and A_2 $(1,2,s \oplus r_{12})$.
Z Gen random r_{13}. Give A_1 $(1,3,r_{13})$ and A_3 $(1,3,s \oplus r_{13})$.
Z Gen random r_{14}. Give A_1 $(1,4,r_{14})$ and A_4 $(1,4,s \oplus r_{14})$.
Z Gen random r_{23}. Give A_2 $(2,3,r_{23})$ and A_3 $(2,3,s \oplus r_{23})$.
Z Gen random r_{24}. Give A_2 $(2,4,r_{24})$ and A_4 $(2,4,s \oplus r_{24})$.
Z Gen random r_{34}. Give A_3 $(3,4,r_{34})$ and A_4 $(3,4,s \oplus r_{34})$.

$(2,4)$-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$
Want A_1, A_2 to determine s, but neither A_1 nor A_2 alone can.
Idea Zelda will secret share with **every pair separately**.
Z Gen random r_{12}. Give A_1 $(1,2,r_{12})$ and A_2 $(1,2,s \oplus r_{12})$.
Z Gen random r_{13}. Give A_1 $(1,3,r_{13})$ and A_3 $(1,3,s \oplus r_{13})$.
Z Gen random r_{14}. Give A_1 $(1,4,r_{14})$ and A_4 $(1,4,s \oplus r_{14})$.
Z Gen random r_{23}. Give A_2 $(2,3,r_{23})$ and A_3 $(2,3,s \oplus r_{23})$.
Z Gen random r_{24}. Give A_2 $(2,4,r_{24})$ and A_4 $(2,4,s \oplus r_{24})$.
Z Gen random r_{34}. Give A_3 $(3,4,r_{34})$ and A_4 $(3,4,s \oplus r_{34})$.
If any two get together they can find secret. No one person can find the secret.

(t, m)-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$.
For each t*-set of* $A_1, \ldots, A_m$ *we set up random strings so they can recover the secret if they all get together.* We omit details but may be on HW.

(t, m)-Secret Sharing via Rand Strings

The secret is $s \in \{0,1\}^n$.
For each t-set of $A_1, \ldots, A_m$ we set up random strings so they can recover the secret if they all get together. We omit details but may be on HW.

Every t-subset does its own secret sharing, so LOTS of strings.

A_i Gets ??? Strings in $(m/2, m)$-Secret Sharing

If do $(m/2, m)$ secret sharing then how many strings does A_1 get?

A_i Gets ??? Strings in $(m/2, m)$-Secret Sharing

If do $(m/2, m)$ secret sharing then how many strings does A_1 get?

A_1 gets a string for every $J \subseteq \{1, \ldots, m\}$, $|J| = \frac{m}{2}$, $1 \in J$.
Equivalent to:

A_i Gets ??? Strings in $(m/2, m)$-Secret Sharing

If do $(m/2, m)$ secret sharing then how many strings does A_1 get?

A_1 gets a string for every $J \subseteq \{1, \ldots, m\}$, $|J| = \frac{m}{2}$, $1 \in J$.
Equivalent to:

A_1 gets a string for every $J \subseteq \{2, \ldots, m\}$, $|J| = \frac{m}{2} - 1$.

A_i Gets ??? Strings in $(m/2, m)$-Secret Sharing

If do $(m/2, m)$ secret sharing then how many strings does A_1 get?

A_1 gets a string for every $J \subseteq \{1, \ldots, m\}$, $|J| = \frac{m}{2}$, $1 \in J$.
Equivalent to:

A_1 gets a string for every $J \subseteq \{2, \ldots, m\}$, $|J| = \frac{m}{2} - 1$.

How many sets? **Discuss**

A_i Gets ??? Strings in $(m/2, m)$-Secret Sharing

If do $(m/2, m)$ secret sharing then how many strings does A_1 get?

A_1 gets a string for every $J \subseteq \{1, \ldots, m\}$, $|J| = \frac{m}{2}$, $1 \in J$.
Equivalent to:

A_1 gets a string for every $J \subseteq \{2, \ldots, m\}$, $|J| = \frac{m}{2} - 1$.

How many sets? **Discuss**

$$\binom{m-1}{\frac{m}{2}-1} \sim \frac{2^m}{\sqrt{m}} \text{ strings}$$

A_i Gets ??? Strings in $(m/2, m)$-Secret Sharing

If do $(m/2, m)$ secret sharing then how many strings does A_1 get?

A_1 gets a string for every $J \subseteq \{1, \ldots, m\}$, $|J| = \frac{m}{2}$, $1 \in J$.
Equivalent to:

A_1 gets a string for every $J \subseteq \{2, \ldots, m\}$, $|J| = \frac{m}{2} - 1$.

How many sets? **Discuss**

$$\binom{m-1}{\frac{m}{2}-1} \sim \frac{2^m}{\sqrt{m}} \text{ strings}$$

Thats A LOT of Strings!

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.
VOTE

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.
VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.
6. $O(\log m)$ strings but not constant.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.
6. $O(\log m)$ strings but not constant.
7. $O(1)$ strings.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.
6. $O(\log m)$ strings but not constant.
7. $O(1)$ strings.

You can always do this with everyone getting 1 string.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.
6. $O(\log m)$ strings but not constant.
7. $O(1)$ strings.

You can always do this with everyone getting 1 string.

I know what you are thinking:

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.
6. $O(\log m)$ strings but not constant.
7. $O(1)$ strings.

You can always do this with everyone getting 1 string.

I know what you are thinking: LOOOONG string.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.

VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.
6. $O(\log m)$ strings but not constant.
7. $O(1)$ strings.

You can always do this with everyone getting 1 string.

I know what you are thinking: LOOOONG string.No.

Reduce The Number of Strings for $(m/2, m)$?

In our $(m/2, m)$-scheme each A_i gets $\sim \frac{2^m}{\sqrt{m}}$ strings.
VOTE

1. Requires roughly 2^m strings.
2. $O(\beta^m)$ strings for some $1 < \beta < 2$ but not poly.
3. $O(m^a)$ strings for some $a > 1$ but not linear.
4. $O(m)$ strings but not m^a with $a < 1$.
5. $O(m^a)$ strings for some $a < 1$ but not logarithmic.
6. $O(\log m)$ strings but not constant.
7. $O(1)$ strings.

You can always do this with everyone getting 1 string.
I know what you are thinking: LOOOONG string.No.
You can always do this with everyone getting 1 string that is the same length as the secret

Convention On Secrets

From now on the secret will always be an element of $\mathbb{Z}_p$ for some primes p.

Convention On Secrets

From now on the secret will always be an element of $\mathbb{Z}_p$ for some primes p.

Example If the secret is 20 then you must operate in $\mathbb{Z}_{23}$.

Convention On Secrets

From now on the secret will always be an element of $\mathbb{Z}_p$ for some primes p.

Example If the secret is 20 then you must operate in $\mathbb{Z}_{23}$.

Always take the smallest prime larger than the secret.

Convention On Secrets

From now on the secret will always be an element of $\mathbb{Z}_p$ for some primes p.

Example If the secret is 20 then you must operate in $\mathbb{Z}_{23}$.

Always take the smallest prime larger than the secret.

If Secret is 23 then take $p = 23$, so now secret is 0.

Secret Sharing With Polynomials: (3,6)

We do $(3, 6)$-Secret Sharing but technique works for any (t, m).

Secret Sharing With Polynomials: (3,6)

We do $(3,6)$-Secret Sharing but technique works for any (t,m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.

Secret Sharing With Polynomials: (3,6)

We do $(3,6)$-Secret Sharing but technique works for any (t, m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand numbers $a_2, a_1 \in \mathbb{Z}_p$

Secret Sharing With Polynomials: (3,6)

We do $(3, 6)$-Secret Sharing but technique works for any (t, m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand numbers $a_2, a_1 \in \mathbb{Z}_p$
3. Zelda forms polynomial $f(x) = a_2x^2 + a_1x + s$.

Secret Sharing With Polynomials: (3,6)

We do $(3, 6)$-Secret Sharing but technique works for any (t, m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand numbers $a_2, a_1 \in \mathbb{Z}_p$
3. Zelda forms polynomial $f(x) = a_2x^2 + a_1x + s$.
4. Zelda gives A_1 $f(1)$, A_2 $f(2)$, ..., A_6 $f(6)$ (all mod p). These are all in $\mathbb{Z}_p$. (Everyone also has p.)

Secret Sharing With Polynomials: (3,6)

We do $(3,6)$-Secret Sharing but technique works for any (t,m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand numbers $a_2, a_1 \in \mathbb{Z}_p$
3. Zelda forms polynomial $f(x) = a_2x^2 + a_1x + s$.
4. Zelda gives A_1 $f(1)$, A_2 $f(2)$, ..., A_6 $f(6)$ (all mod p). These are all in $\mathbb{Z}_p$. (Everyone also has p.)

1. Any 3 have 3 points from $f(x)$ so can find $f(x)$, s.

Secret Sharing With Polynomials: (3,6)

We do $(3,6)$-Secret Sharing but technique works for any (t,m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand numbers $a_2, a_1 \in \mathbb{Z}_p$
3. Zelda forms polynomial $f(x) = a_2x^2 + a_1x + s$.
4. Zelda gives A_1 $f(1)$, A_2 $f(2)$, ..., A_6 $f(6)$ (all mod p). These are all in $\mathbb{Z}_p$. (Everyone also has p.)

1. Any 3 have 3 points from $f(x)$ so can find $f(x)$, s.
2. Any 2 have 2 points from $f(x)$. From these two points what can they conclude?

Secret Sharing With Polynomials: (3,6)

We do $(3, 6)$-Secret Sharing but technique works for any (t, m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand numbers $a_2, a_1 \in \mathbb{Z}_p$
3. Zelda forms polynomial $f(x) = a_2x^2 + a_1x + s$.
4. Zelda gives A_1 $f(1)$, A_2 $f(2)$, ..., A_6 $f(6)$ (all mod p). These are all in $\mathbb{Z}_p$. (Everyone also has p.)

1. Any 3 have 3 points from $f(x)$ so can find $f(x)$, s.
2. Any 2 have 2 points from $f(x)$. From these two points what can they conclude? **NOTHING!**

Secret Sharing With Polynomials: (3,6)

We do $(3,6)$-Secret Sharing but technique works for any (t,m).

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand numbers $a_2, a_1 \in \mathbb{Z}_p$
3. Zelda forms polynomial $f(x) = a_2x^2 + a_1x + s$.
4. Zelda gives A_1 $f(1)$, A_2 $f(2)$, …, A_6 $f(6)$ (all mod p). These are all in $\mathbb{Z}_p$. (Everyone also has p.)

1. Any 3 have 3 points from $f(x)$ so can find $f(x)$, s.
2. Any 2 have 2 points from $f(x)$. From these two points what can they conclude? **NOTHING!** If they know $f(1) = 3$ and $f(2) = 7$ and f is degree 2 then the constant term can be **anything** in $\{0, \ldots, p\}$. So they know NOTHING about s.

Threshold Secret Sharing With Polynomials: (t, m)

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

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

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand $a_{t-1}, \ldots, a_1 \in \mathbb{Z}_p$.

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand $a_{t-1}, \ldots, a_1 \in \mathbb{Z}_p$.
3. Zelda forms polynomial $f(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + s$.

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand $a_{t-1}, \ldots, a_1 \in \mathbb{Z}_p$.
3. Zelda forms polynomial $f(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + s$.
4. For $1 \le i \le m$ Zelda gives A_i $f(i)$ mod p.

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand $a_{t-1}, \ldots, a_1 \in \mathbb{Z}_p$.
3. Zelda forms polynomial $f(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + s$.
4. For $1 \le i \le m$ Zelda gives A_i $f(i)$ mod p.

1. Any t have t points from $f(x)$ so can find $f(x)$, s.

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand $a_{t-1}, \ldots, a_1 \in \mathbb{Z}_p$.
3. Zelda forms polynomial $f(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + s$.
4. For $1 \le i \le m$ Zelda gives A_i $f(i)$ mod p.

1. Any t have t points from $f(x)$ so can find $f(x)$, s.
2. Any $t-1$ have $t-1$ points from $f(x)$. From these $t-1$ points what can they conclude?

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand $a_{t-1}, \ldots, a_1 \in \mathbb{Z}_p$.
3. Zelda forms polynomial $f(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + s$.
4. For $1 \le i \le m$ Zelda gives A_i $f(i)$ mod p.

1. Any t have t points from $f(x)$ so can find $f(x)$, s.
2. Any $t-1$ have $t-1$ points from $f(x)$. From these $t-1$ points what can they conclude? **NOTHING!**

Threshold Secret Sharing With Polynomials: (t, m)

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

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

1. Secret $s \in \mathbb{Z}_p$. Zelda works mod p.
2. Zelda gen rand $a_{t-1}, \ldots, a_1 \in \mathbb{Z}_p$.
3. Zelda forms polynomial $f(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + s$.
4. For $1 \le i \le m$ Zelda gives A_i $f(i)$ mod p.

1. Any t have t points from $f(x)$ so can find $f(x)$, s.
2. Any $t-1$ have $t-1$ points from $f(x)$. From these $t-1$ points what can they conclude? **NOTHING!** Any constant term is consistent with what they know.‘ So they know NOTHING about s.

Example

$(3, 6)$ secret sharing.
$s = 20$ and $p = 37$.

Example

$(3, 6)$ secret sharing.
$s = 20$ and $p = 37$.

1. Zelda picks $a_2 = 8$ and $a_1 = 13$.

Example

$(3, 6)$ secret sharing.
$s = 20$ and $p = 37$.

1. Zelda picks $a_2 = 8$ and $a_1 = 13$.
2. Zelda forms polynomial $f(x) = 8x^2 + 13x + 20$.

Example

$(3, 6)$ secret sharing.
$s = 20$ and $p = 37$.

1. Zelda picks $a_2 = 8$ and $a_1 = 13$.
2. Zelda forms polynomial $f(x) = 8x^2 + 13x + 20$.
3. Zelda gives A_1 $f(1) = 4$, A_2 $f(2) = 4$, A_3 $f(3) = 20$, A_4 $f(4) = 15$, A_5 $f(5) = 26$, A_6 $f(6) = 16$.

Example

$(3, 6)$ secret sharing.
$s = 20$ and $p = 37$.

1. Zelda picks $a_2 = 8$ and $a_1 = 13$.
2. Zelda forms polynomial $f(x) = 8x^2 + 13x + 20$.
3. Zelda gives A_1 $f(1) = 4$, A_2 $f(2) = 4$, A_3 $f(3) = 20$, A_4 $f(4) = 15$, A_5 $f(5) = 26$, A_6 $f(6) = 16$.

If A_1, A_3, A_4 get together and want to find $f(x)$ hence s.
$f(x) = a_2x^2 + a_1x + s$.
$f(1) = 4$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$f(3) = 20$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
$f(4) = 15$: $a_2 \times 4^2 + a_1 \times 4 + s \equiv 14 \pmod{37}$

Example

$(3, 6)$ secret sharing.
$s = 20$ and $p = 37$.

1. Zelda picks $a_2 = 8$ and $a_1 = 13$.
2. Zelda forms polynomial $f(x) = 8x^2 + 13x + 20$.
3. Zelda gives A_1 $f(1) = 4$, A_2 $f(2) = 4$, A_3 $f(3) = 20$, A_4 $f(4) = 15$, A_5 $f(5) = 26$, A_6 $f(6) = 16$.

If A_1, A_3, A_4 get together and want to find $f(x)$ hence s.
$f(x) = a_2x^2 + a_1x + s$.
$f(1) = 4$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$f(3) = 20$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
$f(4) = 15$: $a_2 \times 4^2 + a_1 \times 4 + s \equiv 14 \pmod{37}$
3 linear equations in 3 variables, over mod 37 can be solved.

Example

$(3, 6)$ secret sharing.
$s = 20$ and $p = 37$.

1. Zelda picks $a_2 = 8$ and $a_1 = 13$.
2. Zelda forms polynomial $f(x) = 8x^2 + 13x + 20$.
3. Zelda gives A_1 $f(1) = 4$, A_2 $f(2) = 4$, A_3 $f(3) = 20$, A_4 $f(4) = 15$, A_5 $f(5) = 26$, A_6 $f(6) = 16$.

If A_1, A_3, A_4 get together and want to find $f(x)$ hence s.
$f(x) = a_2x^2 + a_1x + s$.
$f(1) = 4$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$f(3) = 20$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
$f(4) = 15$: $a_2 \times 4^2 + a_1 \times 4 + s \equiv 14 \pmod{37}$
3 linear equations in 3 variables, over mod 37 can be solved.
Note Only need constant term s but can get all coeffs.

What if Two Get Together?

What if A_1 and A_3 get together:
$f(1) = 4$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$f(3) = 20$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
Can they solve these to find s **Discuss**.

What if Two Get Together?

What if A_1 and A_3 get together:
$f(1) = 4$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$f(3) = 20$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
Can they solve these to find s **Discuss**.

No. However, can they use these equations to eliminate some values of s? **Discuss**.

What if Two Get Together?

What if A_1 and A_3 get together:
$f(1) = 4$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$f(3) = 20$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
Can they solve these to find s **Discuss**.

No. However, can they use these equations to eliminate some values of s? **Discuss**.

No. ANY s is consistent. If you pick a value of s, you then have two equations in two variables that can be solved.

What if Two Get Together?

What if A_1 and A_3 get together:
$f(1) = 4$: $a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$f(3) = 20$: $a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
Can they solve these to find s **Discuss**.

No. However, can they use these equations to eliminate some values of s? **Discuss**.

No. ANY s is consistent. If you pick a value of s, you then have two equations in two variables that can be solved.

Important Information-Theoretic Secure: if A_1 and A_3 meet they learn NOTHING. If they had big fancy supercomputers they would still learn NOTHING.

A Note About Linear Equations

The three equations below, over mod 37, can be solved:
$a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
$a_2 \times 4^2 + a_1 \times 4 + s \equiv 15 \pmod{37}$

Could we have solved this had we used mod 32?
VOTE

A Note About Linear Equations

The three equations below, over mod 37, can be solved:

$a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$

$a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$

$a_2 \times 4^2 + a_1 \times 4 + s \equiv 15 \pmod{37}$

Could we have solved this had we used mod 32?

VOTE

1. YES

A Note About Linear Equations

The three equations below, over mod 37, can be solved:

$a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
$a_2 \times 4^2 + a_1 \times 4 + s \equiv 15 \pmod{37}$

Could we have solved this had we used mod 32?
VOTE

1. YES
2. NO

A Note About Linear Equations

The three equations below, over mod 37, can be solved:
$a_2 \times 1^2 + a_1 \times 1 + s \equiv 4 \pmod{37}$
$a_2 \times 3^2 + a_1 \times 3 + s \equiv 20 \pmod{37}$
$a_2 \times 4^2 + a_1 \times 4 + s \equiv 15 \pmod{37}$

Could we have solved this had we used mod 32?
VOTE

1. YES
2. NO

These equations, Don't know, but in general, NO
Need a domain where every number has a mult inverse.
Over mod p, p primes, all numbers have mult inverses.
Over mod 32, even numbers do not have mult inverse.

Threshold Secret Sharing With Polynomials: Ref

Due to Adi Shamir

How to Share a Secret

Communication of the ACM

Volume 22, Number 11

1979

We Used Polynomials. Could Use. . .

What did we use about degree $t-1$ polynomials?

We Used Polynomials. Could Use...

What did we use about degree $t-1$ polynomials?

1. t points determine the polynomial (we need constant term).

We Used Polynomials. Could Use...

What did we use about degree $t-1$ polynomials?

1. t points determine the polynomial (we need constant term).
2. $t-1$ points give **no information** about constant term.

We Used Polynomials. Could Use...

What did we use about degree $t-1$ polynomials?

1. t points determine the polynomial (we need constant term).
2. $t-1$ points give **no information** about constant term.

Could do geometry over $\mathbb{Z}_p^3$. A **Plane** in $\mathbb{Z}_p^3$ is:

$$\{(x,y,z) : ax+by+cz=d\}$$

We Used Polynomials. Could Use...

What did we use about degree $t-1$ polynomials?

1. t points determine the polynomial (we need constant term).
2. $t-1$ points give **no information** about constant term.

Could do geometry over $\mathbb{Z}_p^3$. A **Plane** in $\mathbb{Z}_p^3$ is:

$$\{(x,y,z) : ax+by+cz=d\}$$

1. 3 points in $\mathbb{Z}_p^3$ determine a plane.

We Used Polynomials. Could Use...

What did we use about degree $t-1$ polynomials?

1. t points determine the polynomial (we need constant term).
2. $t-1$ points give **no information** about constant term.

Could do geometry over $\mathbb{Z}_p^3$. A **Plane** in $\mathbb{Z}_p^3$ is:

$$\{(x,y,z) : ax+by+cz=d\}$$

1. 3 points in $\mathbb{Z}_p^3$ determine a plane.
2. 2 points in $\mathbb{Z}_p^3$ give **no information** about d.

This approach is due to George Blakely, **Safeguarding Cryptographic Keys**, **International Workshop on Managing Requirements**, **Vol 48**, **1979**.
We will not do secret sharing this way, though one could.

We Used Polynomials. Could Use...

We won't go into details but there are two ways to use the **Chinese Remainder Thm** to do Secret Sharing.

Due to:
C.A. Asmuth and J. Bloom. **A modular approach to key safeguarding. IEEE Transactions on Information Theory Vol 29**, **Number 2**, **208-210**, **1983**.

And Independently

M. Mignotte **How to share a secret**, **Cryptography: Proceedings of the Workshop on Cryptography, Burg Deursetein**, **Volume 149 of Lecture Notes in Computer Science**, **1982**.

Features and Caveats of Poly Method

Imagine that you've done (t, m) secret sharing with polynomial, $p(x)$. So for $1 \leq i \leq m$, A_i has $f(i)$.

Features and Caveats of Poly Method

Imagine that you've done (t, m) secret sharing with polynomial, $p(x)$. So for $1 \le i \le m$, A_i has $f(i)$.

1. **Feature** If more people come FINE- can extend to $(t, m+a)$ by giving A_{m+1}, $f(m+1)$, ..., A_{m+a}, $f(m+a)$.

Features and Caveats of Poly Method

Imagine that you've done (t, m) secret sharing with polynomial, $p(x)$. So for $1 \le i \le m$, A_i has $f(i)$.

1. **Feature** If more people come FINE- can extend to $(t, m+a)$ by giving A_{m+1}, $f(m+1)$, ..., A_{m+a}, $f(m+a)$.
2. **Caveat** If $m \ge p$ then you run out of points to give people. There are ways to deal with this, but we will not bother. We will always assume $m < p$.

BILL STOP
RECORDING LECTURE