Threshold Secret Sharing: Shorter Shares via Comp Security

Ideal Secret Sharing: Info-Theoretic
Zelda has a secret $s \in \{0, 1\}^n$.

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

1. If any $t$ get together than they can learn the secret.
2. If any $t - 1$ get together they cannot learn the secret.

**Cannot learn the secret** Last lecture this was Info-Theoretic. This lecture we consider info-theoretic and comp-theoretic.
Info-Theoretic: Shares are $\geq n$

Info-theoretic $(t, L)$-Secret Sharing.
If $A_t$ has a share of length $n - 1$ then $A_1, \ldots, A_{t-1}$ CAN learn something (so NOT info-theoretic security).

$A_1, \ldots, A_{t-1}$ do the following:

$\text{CAND} = \emptyset$. $\text{CAND}$ will be set of Candidates for $s$.
For $x \in \{0, 1\}^{n-1}$ (go through ALL shares $A_t$ could have)

$A_1, \ldots, A_{t-1}$ pretend $A_t$ has $x$ and deduce candidates secret $s'$

$\text{CAND} := \text{CAND} \cup \{s'\}$

Secret is in $\text{CAND}$. $|\text{CAND}| = 2^{n-1} < 2^n$. So we have eliminated many strings from being the $s$
Are Shorter Shares Ever 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 $< \beta n$ with a hardness assumption.
2. Even with hardness assumption REQUIRES shares $\geq n$. 
Are Shorter Shares Ever 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 $< \beta n$ with a hardness assumption.
2. Even with hardness assumption REQUIRES shares $\geq n$.

**Can get shares $< \beta n$ with a hardness assumption.**

Will do that later.
Threshold Secret Sharing: Computational
Recall

Threshold Secret Sharing: Information-Theoretic

1. Secret is \( s \in \{0, 1\}^n \).
2. \((t, L)\): \( t \) people can find \( s \), but \( t - 1 \) cannot.
3. There is a \((t, L)\)-scheme where all gets a share of size \( n \).
4. There is no scheme where someone gets a share of size \( < n \).

That is for Information-Theoretic Security.

What if we settle for Computational Security.
Recall

Threshold Secret Sharing: Information-Theoretic

1. Secret is $s \in \{0, 1\}^n$.
2. $(t, L)$: $t$ people can find $s$, but $t - 1$ cannot.
3. There is a $(t, L)$-scheme where all gets a share of size $n$.
4. There is no scheme where someone gets a share of size $< n$.

That is for **Information-Theoretic Security**.

What if we settle for **Computational Security**.

**Promise to you:** No more **Punking**
For ciphertext only:

1. Shift is crackable if text is long
2. Affine is crackable if text is long
3. Vig is crackable if text is long compared to the key
4. Matrix is crackable if text is long compared to the key
   (actually I do not know if this is true)

Is there an encryption system where the $|k| < |T|$ and the system is computationally secure?

Need to define terms first.
**Def:** Let $0 < \alpha \leq 1$. An \textbf{\(\alpha\)-Symmetric Encryption System (\(\alpha\)-SES)} is a three tuple $(GEN, ENC, DEC)$ where

1. $GEN$ takes $n$ and generates $k \in \{0, 1\}^{\alpha n}$.

2. $ENC$ takes $k \in \{0, 1\}^{\alpha n}$ and $m \in \{0, 1\}^n$, outputs $c \in \{0, 1\}^n$. ($ENC$ encrypts $m$ with key $k$. We denote $ENC_k(m)$).

3. $DEC$ takes $k \in \{0, 1\}^{\alpha n}$ and $c \in \{0, 1\}^n$ and outputs $m \in \{0, 1\}^n$ such that $DEC_k(ENC_k(m)) = m$.

**Def:** We will not define security formally here; however, intuitively Eve cannot learn $m$ from $c$. We are concerned with ciphertext only.

**Note:** $\alpha$-SES encrypts length $n$ message with length $n$ ciphertext.
Psuedorandom Generators

**Def:** (Informal) A pseudorandom gen maps a short seed to a long sequence that a limited Eve cannot distinguish from random.

**Idea:** Do the one-time-pad but with a pseudorandom sequence.

Discuss

**PROS** and **CONS**
Psuedorandom Generators

**Def:** (Informal) A pseudorandom gen maps a short seed to a long sequence that a limited Eve cannot distinguish from random.

**Idea:** Do the one-time-pad but with a psuedorandom sequence.

**Discuss**

**PROS** and **CONS**

**CON:** All Powerful Even can crack it!

**PRO:** Limited Eve cannot crack it!
Psuedorandom Generators

**Def:** (Informal) A pseudorandom gen maps a short seed to a long sequence that a limited Eve cannot distinguish from random.

**Idea:** Do the one-time-pad but with a psuedorandom sequence.

**Discuss**

**PROS** and **CONS**

**CON:** All Powerful Even can crack it!

**PRO:** Limited Eve cannot crack it!
Psuedorandom Generators

Def: (Informal) A pseudorandom gen maps a short seed to a long sequence that a limited Eve cannot distinguish from random.

Idea: Do the one-time-pad but with a pseudorandom sequence.

Discuss

PROS and CONS

CON: All Powerful Even can crack it!
PRO: Limited Eve cannot crack it!
PRO: Can Actually use!
BBS Generator

Blum-Blum-Shub psuedo-random Generator:

1. Seed: $p, q$ primes, $x_0 \in \mathbb{Z}_{N=pq}$. $p, q \equiv 3 \pmod{4}$,
2. Sequence:

\[ x_1 = x_0^2 \mod N \quad b_1 = \text{LSB}(x_1) \]
\[ x_2 = x_1^2 \mod N \quad b_2 = \text{LSB}(x_2) \]
\[ \vdots \quad \vdots \quad \vdots \]
\[ x_L = x_{L-1}^2 \mod N \quad b_L = \text{LSB}(x_L) \]

\[ r = b_1 \cdots b_L \] is psuedo-random.

**Known:** Assume determining if a number is in $SQ_N$ is hard. If $L$ is twice the length of seed, and seed long, enough then secure.
Example of $\frac{1}{2}$-SES

1. **GEN:** $k = (p, q, x_0)$. $|k| = \frac{n}{2}$. $p, q$ prime $p \equiv q \equiv 3 \pmod{4}$.

2. **ENC:** Use $k$ to BBS-gen $b_1, \ldots, b_n$. $m \in \{0, 1\}^n$.

   $$ENC_k(m_1, \ldots, m_n) = (m_1 \oplus b_1, \ldots, m_n \oplus b_n).$$

3. **DEC:** Bob can use $k = (p, q, x_0)$ to find $b_0, \ldots, b_n$, and decode.

**Known:** Assume determining if a number is in $SQ_N$ is hard. For large enough $n$ this is secure.

**Note:** Message is twice as long as key, so this is $\frac{1}{2}$-SES.

**Note:** Will not be using this particular SES but have it here as a concrete example.
Short Shares

**Thm:** Assume there exists an $\alpha$-SES. Assume that for message of length $n$, it is secure. Then, for all $1 \leq t \leq L$ there is a $(t, L)$-scheme for $|s| = n$ where each share is of size $\frac{n}{t} + \alpha n$.

1. Zelda does $k \leftarrow \text{GEN}(n)$. Note $|k| = \alpha n$.
2. $u = \text{ENC}_k(s)$. Let $u = u_0 \cdots u_{t-1}$, $|u_i| \sim \frac{n}{t}$.
3. Let $p \sim 2^{n/t}$. Zelda forms poly over $\mathbb{Z}_p$:

   $$f(x) = u_{t-1}x^{t-1} + \cdots + u_1x + u_0$$

4. Let $q \sim 2^{\alpha n}$. Zelda forms poly over $\mathbb{Z}_q$ by choosing $r_{t-1}, \ldots, r_1 \in \{0, \ldots, q - 1\}$ at random and then:

   $$g(x) = r_{t-1}x^{t-1} + \cdots + r_1x + k$$

5. Zelda gives $A_i, (f(i), g(i))$. Length: $\sim \frac{n}{t} + \alpha n$. 
Length and Recovery

**Length:**
1. $f(i) \in \mathbb{Z}_p$ where $p \sim 2^{n/t}$, so $|f(i)| \sim \frac{n}{t}$.
2. $g(i) \in \mathbb{Z}_q$ where $q \sim 2^{\alpha n}$, so $|g(i)| \sim \alpha n$.

**Recovery:** If $t$ get together:
1. Have $t$ points of $f$, can get $u_{t-1}, \ldots, u_0$, hence $u$.
2. $u = ENC_k(s)$. So need $k$.
3. Have $t$ points of $g$, can get $k$.
4. With $k$ and $u$ can get $s = DEC_k(u)$. 
Length and Recovery

**Length:**

1. $f(i) \in \mathbb{Z}_p$ where $p \sim 2^{n/t}$, so $|f(i)| \sim \frac{n}{t}$.
2. $g(i) \in \mathbb{Z}_q$ where $q \sim 2^{\alpha n}$, so $|g(i)| \sim \alpha n$.

**Recovery:** If $t$ get together:

1. Have $t$ points of $f$, can get $u_{t-1}, \ldots, u_0$, hence $u$.
2. $u = ENC_k(s)$. So need $k$.
3. Have $t$ points of $g$, can get $k$.
4. With $k$ and $u$ can get $s = DEC_k(u)$.

If $t - 1$ get together:
Length and Recovery

Length:
1. \( f(i) \in \mathbb{Z}_p \) where \( p \sim 2^{n/t} \), so \( |f(i)| \sim \frac{n}{t} \).
2. \( g(i) \in \mathbb{Z}_q \) where \( q \sim 2^{\alpha n} \), so \( |g(i)| \sim \alpha n \).

Recovery: If \( t \) get together:
1. Have \( t \) points of \( f \), can get \( u_{t-1}, \ldots, u_0 \), hence \( u \).
2. \( u = ENC_k(s) \). So need \( k \).
3. Have \( t \) points of \( g \), can get \( k \).
4. With \( k \) and \( u \) can get \( s = DEC_k(u) \).

If \( t - 1 \) get together:
Next Slide
The scheme I showed you is due to Hugo Krawczyk, *Secret Sharing Made Short*, *Advances in Crypto – CRYPTO 1993 Lecture notes in computer science 773*, 1993

However, the proof of security was not quite right.

Mihir Bellar and Phillip Rogaway wrote a paper that proved Krawczyk’s protocol secure by adding a condition to the $\alpha$-SES. We omit since its complicated.

Can we do better than $\frac{n}{t} + \alpha n$?

**Ill Formed Question:** Can we do better than $\frac{n}{t} + \alpha n$?
The question is not quite right – if we have a smaller $\alpha$ can do better.

**Better Question:** Assume there is an $\alpha$-SES. Is the following true:

> For all $0 < \beta < 1$ there exists an $(t, L)$ secret sharing scheme where everyone gets $\frac{n}{t} + \beta n$.

**Discuss**
Can we do better than $\frac{n}{t} + \alpha n$?

**Ill Formed Question:** Can we do better than $\frac{n}{t} + \alpha n$?
The question is not quite right – if we have a smaller $\alpha$ can do better.

**Better Question:** Assume there is an $\alpha$-SES. Is the following true:

*For all $0 < \beta < 1$ there exists an $(t, L)$ secret sharing scheme where everyone gets $\frac{n}{t} + \beta n$.*

**Discuss**

Can be done by iterating the above construction. Might be HW or Exam.
(2, 2): $A, B$ share the secret $s$, $|s| = n$. Computational Secret Sharing, so can make a hardness assumption.

**Question:** Is there a (2, 2) secret sharing scheme where $A$ and $B$ both get a share $\leq \frac{n}{3}$?

**Discuss.** Vote!

1. YES! There is such a Scheme.
2. NO! We can prove there is NO such scheme.
3. PUNKED! Bill will shows us a scheme that looks like it works but he’ll be PUNKING US!
4. Unknown to science!
Breaking the $\frac{n}{t}$ Barrier!

(2, 2): $A, B$ share the secret $s$, $|s| = n$. Computational Secret Sharing, so can make a hardness assumption.

**Question:** Is there a (2, 2) secret sharing scheme where $A$ and $B$ both get a share $\leq \frac{n}{3}$? 

**Discuss.** Vote!

1. YES! There is such a Scheme.
2. NO! We can prove there is NO such scheme.
3. PUNKED! Bill will shows us a scheme that looks like it works but he’ll be PUNKING US!
4. Unknown to science!

NO! We can prove there is NO such scheme.
Can’t Break the $\frac{n}{t}$ Barrier!

**Theorem:** There is no $(2, 2)$-scheme with shares $\frac{n}{3}$.

**Proof:** Assume there is.
Map $s \in \{0, 1\}^n$ to the ordered pair ($A$’s share, $B$’s share)
$2^n$ elements in the domain.
$2^{n/3} \times 2^{n/3} = 2^{2n/3}$ elements in the co-domain.

Hence exists $s, s' \in \{0, 1\}^n$ that map to same $(a, b)$.
If $A$ gets $a$, and $B$ gets $b$, will not decode uniquely into one secret.

**Contradiction!**
This Generalizes. Might be on HW or Exam
Ideal Secret Sharing: Shares of Length “Exactly” $n$
Ideal Secret Sharing

Definition
A sec. sharing sch. is **ideal** if all shares same size as secret.

Definition
An **Access Structure** is a subset of \( \{A_1, \ldots, A_k\} \) closed under superset. \((t, L)\) is **threshold access structure**.

We have shown that the threshold access structures has an ideal sec. sharing scheme with poly method.

Do other access structures have ideal schemes?
An non-Th Access Structure with Ideal Sec Sharing

$A, B_1, B_2, C_1, C_2$. Want $A \land B_1 \land B_2 \lor A \land C_1 \land C_2$ to get $s$.

1. Zelda picks rand $r \in \{0, 1\}^n$, gives to $A$.
2. Zelda computes $s' = r \oplus s$.
3. Zelda does $(2, 2)$ with $s' = s \oplus r$ for $B_1, B_2$.
4. Zelda does $(2, 2)$ with $s' = s \oplus r$ for $C_1, C_2$.

1. $A, B_1, B_2$ have $r$ and $r \oplus s$, can get $s$.
2. $A, C_1, C_2$ have $r$ and $r \oplus s$, can get $s$.
3. Any superset of $\{A, B_1, B_2\}$ or $\{A, C_1, C_2\}$ can get $s$.
4. Any other set just has some random strings.
1. Threshold Secret sharing: if $t$ or more get together.
2. Let $G$ be a graph. Let $s,t$ be nodes. People are edges. Any connected path can get the secret.
3. Monotone Boolean Formulas where each variable occurs once. Example:
   \[ A \land ((B_1 \land B_2) \lor (C_1 \land C_2)) \]
4. Monotone Span Programs (Omitted – Matrix Thing)
Access Structures that do not admit Scheme with Share Length $n$

1. $(A_1 \land A_2) \lor (A_2 \land A_3) \lor (A_3 \land A_4)$
2. $(A_1 \land A_2 \land A_3) \lor (A_1 \land A_4) \lor (A_2 \land A_4) \lor (A_3 \land A_4)$ (Captain and Crew. $A_1, A_2, A_3$ is the crew, and $A_4$ is the captain. Entire crew, or captain and 1 crew, can get $s$.
3. $(A_1 \land A_2 \land A_3) \lor (A_1 \land A_4) \lor (A_2 \land A_4)$ (Captain and Rival. $A_1, A_2, A_3$ is the crew, $A_3$ is a rival, $A_4$ is the captain. Entire crew, or captain and 1 crew who is NOT rival, can get $s$.
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.
**Thm:** 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... and 1. It is very **funky**
Open Question

Determine for every 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.
GO OVER HW 07