Revisiting Beimel-Weinreb Weighted Threshold Secret Sharing Schemes

Computer Security Foundations Symposium 2026, Lisboa

Oriol Farràs — **Miquel Guiot**

Universitat Rovira i Virgili

28 July 2026

Introduction: Secret Sharing Schemes

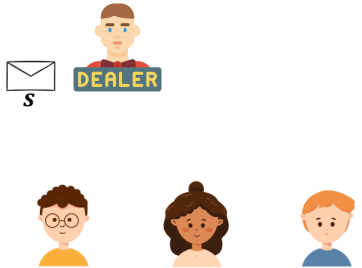
Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Security Requirements

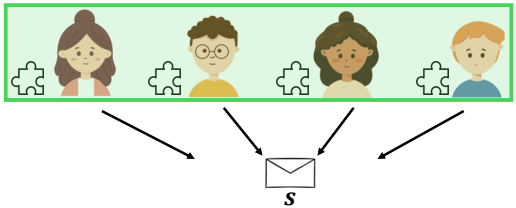
Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Security Requirements

- Correctness



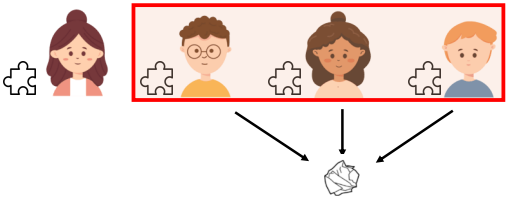
Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Security Requirements

- Correctness
- Privacy



Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Applications

- Secure Multiparty Computation

Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Applications

- Secure Multiparty Computation
- Attribute-based Encryption

Introduction: Secret Sharing Schemes

Secret Sharing Scheme

Cryptographic primitive that allows a dealer to **share** a secret among a set of parties, so that only **authorized** subsets of them can **recover** it.

Applications

- Secure Multiparty Computation
- Attribute-based Encryption
- Weighted Cryptography

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Example

$\Gamma = \{\{$  ,  ,  ,  $\}\}$

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Example

$\Gamma = \{\{$  $,$  $,$  $\},$

$\{$  $,$  $,$  $,$  $\}\}$

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Remarks

- Access structures are monotone.

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Remarks

- Access structures are monotone.
- We consider perfect access structures.

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Remarks

- Access structures are monotone.
- We consider perfect access structures.
- Access structures are expressed as monotone Boolean functions $f : \{0,1\}^n \to \{0,1\}$.

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Example

$\Gamma = \{\{$  ,  ,  ,  $\}\}$

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Example

$$\Gamma = \{\{$$  $$,$$  $$,$$  $$,$$  $$\}\}$$

$$f($$  $$,$$  $$,$$  $$,$$  $$) = 1$$

Introduction: Access Structures

Access Structure

Family of **authorized subsets** of a secret sharing scheme. It is denoted by Γ.

Example

$$\Gamma = \{\{\text{  }, \text{  }, \text{  }, \text{  }\}\}$$

$$f(\text{  }, \text{  }, \text{  }) = 0$$

Introduction: Weighted Cryptography

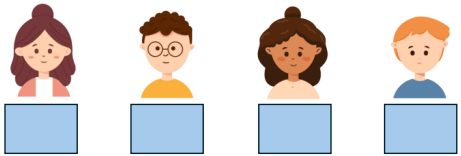
Problem

Traditional cryptosystems assume that all parties are equal.

Introduction: Weighted Cryptography

Problem

Traditional cryptosystems assume that all parties are equal.



Introduction: Weighted Cryptography

Problem

Traditional cryptosystems assume that all parties are equal.

Weighted Cryptography

Each party has a **weight**, and security depends on the **combined weight** of the parties.

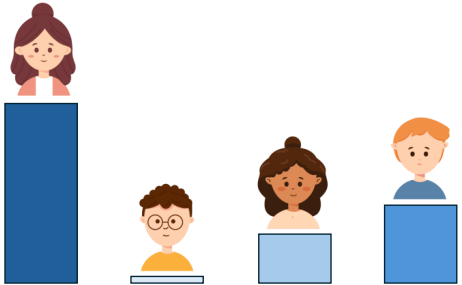
Introduction: Weighted Cryptography

Problem

Traditional cryptosystems assume that all parties are equal.

Weighted Cryptography

Each party has a **weight**, and security depends on the **combined weight** of the parties.



Introduction: Weighted Cryptography

Problem

Traditional cryptosystems assume that all parties are equal.

Weighted Cryptography

Each party has a **weight**, and security depends on the **combined weight** of the parties.

Applications

- Proof-of-Stake Blockchains

Introduction: Weighted Cryptography

Problem

Traditional cryptosystems assume that all parties are equal.

Weighted Cryptography

Each party has a **weight**, and security depends on the **combined weight** of the parties.

Applications

- Proof-of-Stake Blockchains
- Oracle Networks

Introduction: Weighted Cryptography

Problem

Traditional cryptosystems assume that all parties are equal.

Weighted Cryptography

Each party has a **weight**, and security depends on the **combined weight** of the parties.

Applications

- Proof-of-Stake Blockchains
- Oracle Networks
- Verifiable Unpredictable Functions

Introduction: Shamir Secret Sharing Scheme

Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \mathrm{sign}(x_1 + \ldots + x_n - t)}$.

Introduction: Shamir Secret Sharing Scheme

Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \mathrm{sign}(x_1 + \ldots + x_n - t)}$.

Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Introduction: Shamir Secret Sharing Scheme

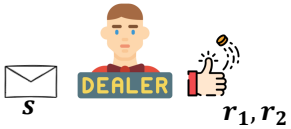
Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + \ldots + x_n - t)}$.

Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$



Introduction: Shamir Secret Sharing Scheme

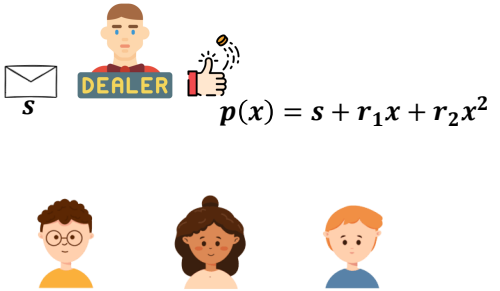
Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + \ldots + x_n - t)}$.

Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$



Introduction: Shamir Secret Sharing Scheme

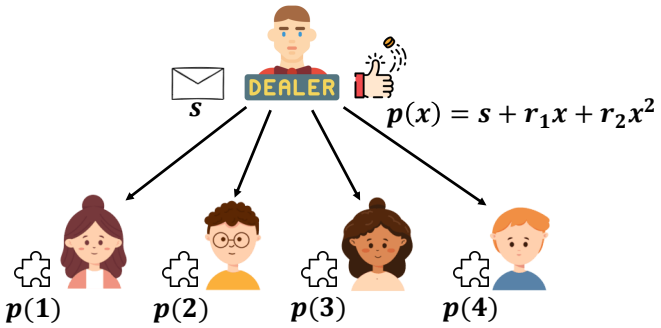
Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + \ldots + x_n - t)}$.

Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$



Introduction: Shamir Secret Sharing Scheme

Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + \ldots + x_n - t)}$.

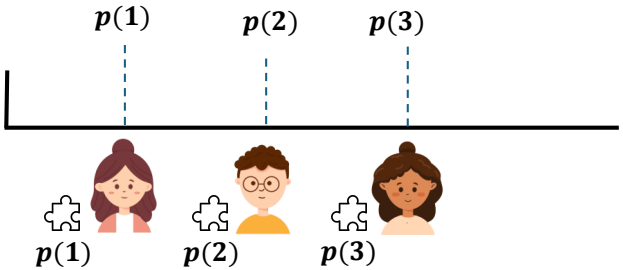
Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$

Security Requirements

- Correctness (≥ 3)



Introduction: Shamir Secret Sharing Scheme

Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + \ldots + x_n - t)}$.

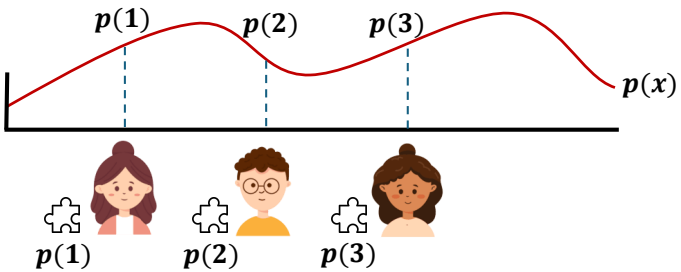
Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$

Security Requirements

- Correctness (≥ 3)



Introduction: Shamir Secret Sharing Scheme

Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + \ldots + x_n - t)}$.

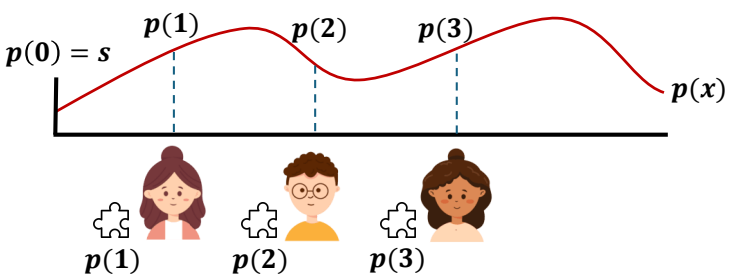
Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$

Security Requirements

- Correctness (≥ 3)



Introduction: Shamir Secret Sharing Scheme

Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + \ldots + x_n - t)}$.

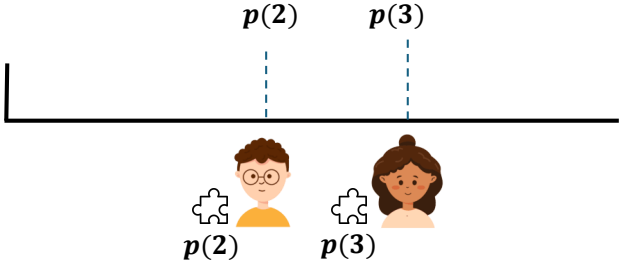
Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$

Security Requirements

- Correctness (≥ 3)
- Privacy (< 3)



Introduction: Shamir Secret Sharing Scheme

Threshold Access Structure

A subset X is authorized if and only if $|X| \geq t$. Γ is given by $\boldsymbol{f(x) = \text{sign}(x_1 + ... + x_n - t)}$.

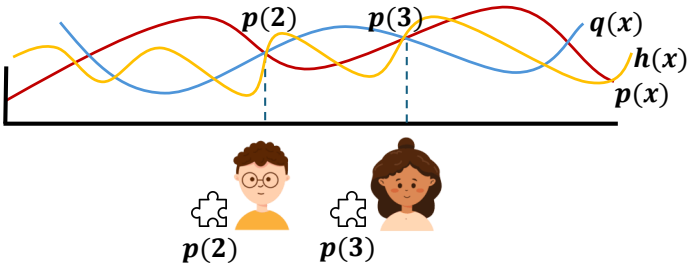
Theorem [Shamir'79]

The threshold access structure admits a secret sharing scheme with share size $\boldsymbol{O(\log n)}$.

Example with $t = 3$

Security Requirements

- Correctness (≥ 3)
- Privacy (< 3)



Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

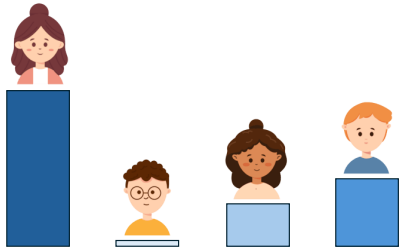
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

Example



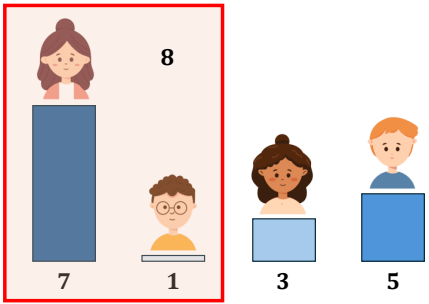
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

Example



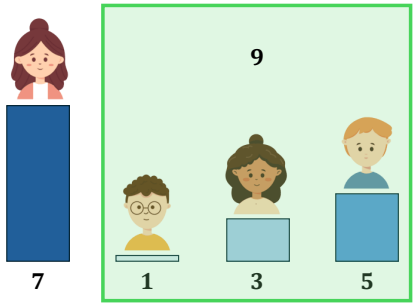
Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

Example



Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

Theorem [Shamir'79]

WTASs admit schemes with share size $\boldsymbol{O(w_i \log W)}$.

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

Theorem [Shamir'79]

WTASs admit schemes with share size $\boldsymbol{O(w_i \log W)}$.

Idea: Mimic Shamir's scheme and give to each party as many points as its weight.

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

Theorem [Shamir'79]

WTASs admit schemes with share size $\boldsymbol{O(w_i \log W)}$.

Idea: Mimic Shamir's scheme and give to each party as many points as its weight.

Problem [Håstadt'94]

There exists a WTAS requiring weights of exponential size.

Introduction: Weighted Threshold Access Structures

Weighted Threshold Access Structure (WTAS)

Each party i has a **weight** w_i and Γ is given by a **Weighted Threshold Function (WTF):**

$$f(x) = \text{sign}(\sum_{i \in [n]} w_i x_i - \sigma)$$

Theorem [Shamir'79]

WTASs admit schemes with share size $\boldsymbol{O(w_i \log W)}$.

Idea: Mimic Shamir's scheme and give to each party as many points as its weight.

Problem [Håstadt'94]

There exists a WTAS requiring weights of exponential size.

- Shamir scheme leads to exponential share size!

Introduction: Weighted Threshold State-of-the-Art

Question: Is there any alternative construction that reduces the share size?

Introduction: Weighted Threshold State-of-the-Art

Question: Is there any alternative construction that reduces the share size?

Theorem [Beimel, Weinreb'06]

WTASs admit schemes with

Introduction: Weighted Threshold State-of-the-Art

Question: Is there any alternative construction that reduces the share size?

Theorem [Beimel, Weinreb'06]

WTASs admit schemes with

- **information-theoretic security** and share size $n^{O(\log n)}$.

Introduction: Weighted Threshold State-of-the-Art

Question: Is there any alternative construction that reduces the share size?

Theorem [Beimel, Weinreb'06]

WTASs admit schemes with

- **information-theoretic security** and share size $n^{O(\log n)}$.
- **computational security**, share size λ and public information size $O(\lambda n^2 \log \sigma)$.

Introduction: Weighted Threshold State-of-the-Art

Question: Is there any alternative construction that reduces the share size?

Theorem [Beimel, Weinreb'06]

WTASs admit schemes with

- **information-theoretic security** and share size $n^{O(\log n)}$.
- **computational security**, share size λ and public information size $O(\lambda n^2 \log \sigma)$.

Idea: To build a monotone Boolean circuit of polynomial size and logarithmic depth.

Introduction: Weighted Threshold State-of-the-Art

Question: Is there any alternative construction that reduces the share size?

Theorem [Beimel, Weinreb'06]

WTASs admit schemes with

- **information-theoretic security** and share size $n^{O(\log n)}$.
- **computational security**, share size λ and public information size $O(\lambda n^2 \log \sigma)$.

Idea: To build a monotone Boolean circuit of polynomial size and logarithmic depth.

Efficiency

For nearly 20 years, these have been the most efficient constructions.

Introduction: Weighted Threshold State-of-the-Art

Question: Is there any alternative construction that reduces the share size?

Theorem [Beimel, Weinreb'06]

WTASs admit schemes with

- **information-theoretic security** and share size $n^{O(\log n)}$.
- **computational security**, share size λ and public information size $\sim 1830\lambda n^2 \log \sigma$.

Idea: To build a monotone Boolean circuit of polynomial size and logarithmic depth.

Efficiency

For nearly 20 years, these have been the most efficient constructions.

Problem

The circuit requires sorting gates that are not feasible in practice.

Introduction: Goal

Question: Can we **improve** the state-of-the-art **weighted threshold** secret sharing schemes?

Introduction: Our Results

This Work

- A new **monotone circuit** for **WTF**
- An **efficient** computational scheme for **WTAS**
- An **information-theoretic** scheme for **ramp WTAS**

Introduction: Our Results

This Work

- A new **monotone circuit** for **WTF**
- An **efficient** computational scheme for **WTAS**
- An **information-theoretic** scheme for **ramp WTAS**

} This Talk

Introduction: Our Results

This Work

- A new **monotone circuit** for **WTF**
- An **efficient** computational scheme for **WTAS**
- An **information-theoretic** scheme for **ramp WTAS**

} This Talk

Main Result: **Computational scheme for WTAS**

- Share size $\boldsymbol{\lambda}$
- Public information size $\boldsymbol{4\lambda n^2 \log \sigma}$
- Usage of only $\boldsymbol{4\lambda n \log \sigma}$ bits of the public information during reconstruction

Introduction: Our Results

This Work

- A new **monotone circuit** for **WTF**
- An **efficient** computational scheme for **WTAS**
- An **information-theoretic** scheme for **ramp WTAS**

} This Talk

Main Result: **Computational scheme for WTAS**

- Share size $\boldsymbol{\lambda}$
- Public information size $\boldsymbol{4\lambda n^2 \log \sigma}$
- Usage of only $\boldsymbol{4\lambda n \log \sigma}$ bits of the public information during reconstruction

Comparison

$300\times - 6700\times$ smaller shares than the previous schemes for deployed blockchain networks.

Revisiting Beimel & Weinreb Schemes: Their Construction

Theorem [Beimel, Weinreb'06]

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\sim \mathbf{1830}\boldsymbol{\lambda n^2 \log \sigma}$

Revisiting Beimel & Weinreb Schemes: Their Construction

Theorem [Beimel, Weinreb'06]

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\sim \mathbf{1830}\boldsymbol{\lambda n^2 \log \sigma}$

Main Steps

$$f(x)$$

Revisiting Beimel & Weinreb Schemes: Their Construction

Theorem [Beimel, Weinreb'06]

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\sim \mathbf{1830}\boldsymbol{\lambda n^2 \log \sigma}$

Main Steps

1. **Reduction** from any WTF to a **universal** WTF with weights **powers of 2**.

$$f(x) \rightarrow f_u(x)$$

Revisiting Beimel & Weinreb Schemes: Their Construction

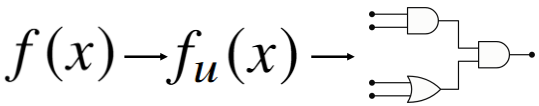
Theorem [Beimel, Weinreb'06]

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\sim \mathbf{1830}\boldsymbol{\lambda n^2 \log \sigma}$

Main Steps

1. **Reduction** from any WTF to a **universal** WTF with weights **powers of 2**.
2. Construct a monotone Boolean circuit of **polynomial size and depth**.



Revisiting Beimel & Weinreb Schemes: Their Construction

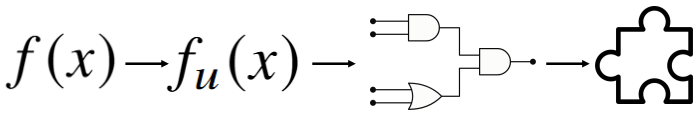
Theorem [Beimel, Weinreb'06]

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\sim \mathbf{1830}\boldsymbol{\lambda} \boldsymbol{n}^{\mathbf{2}} \log \boldsymbol{\sigma}$

Main Steps

1. **Reduction** from any WTF to a **universal** WTF with weights **powers of 2**.
2. Construct a monotone Boolean circuit of **polynomial size and depth**.
3. Apply **Yao's compiler** to obtain the **computational** scheme.



Revisiting Beimel & Weinreb Schemes: Their Construction

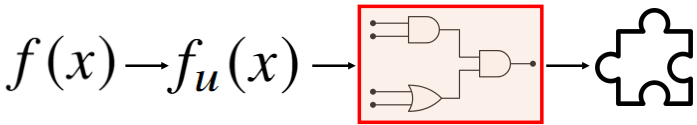
Theorem [Beimel, Weinreb'06]

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\sim \mathbf{1830}\lambda n^2 \log \sigma$

Main Steps

1. **Reduction** from any WTF to a **universal** WTF with weights **powers of 2**.
2. Construct a monotone Boolean circuit of **polynomial size and depth**.
3. Apply **Yao's compiler** to obtain the **computational** scheme.



Revisiting Beimel & Weinreb Schemes: Our Circuit

$$f(x) = \text{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

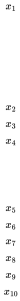
Main Steps

Revisiting Beimel & Weinreb Schemes: Our Circuit

$$f(x) = \mathrm{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Main Steps

1. Set as many **levels** as **distinct** powers of two in the WTF.

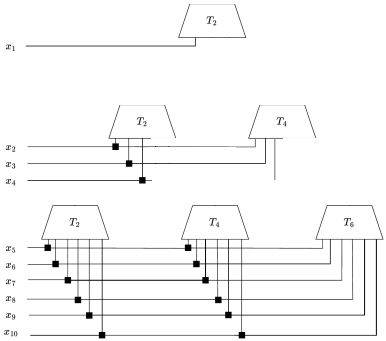


Revisiting Beimel & Weinreb Schemes: Our Circuit

$$f(x) = \mathrm{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Main Steps

1. Set as many **levels** as **distinct** powers of two in the WTF.
2. Use **threshold gates** at each level to **add** the variables and carry bits from previous level.

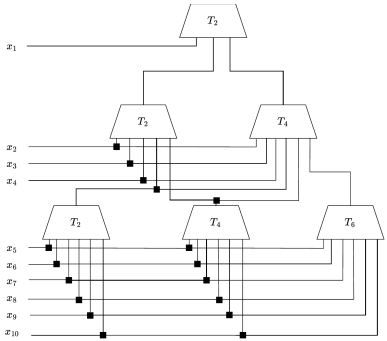


Revisiting Beimel & Weinreb Schemes: Our Circuit

$$f(x) = \mathrm{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Main Steps

1. Set as many **levels** as **distinct** powers of two in the WTF.
2. Use **threshold gates** at each level to **add** the variables and carry bits from previous level.
3. The result acts as the carry bits for the next level by **appropriate wiring**.

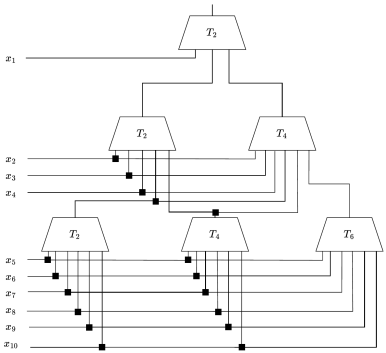


Revisiting Beimel & Weinreb Schemes: Our Circuit

$$f(x) = \mathrm{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Main Steps

1. Set as many **levels** as **distinct** powers of two in the WTF.
2. Use **threshold gates** at each level to **add** the variables and carry bits from previous level.
3. The result acts as the carry bits for the next level by **appropriate wiring**.
4. **Accept** if the carry from the top level is **1**.

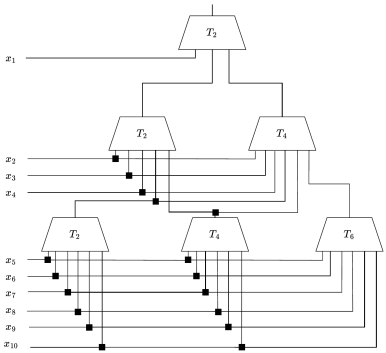


Revisiting Beimel & Weinreb Schemes: Our Circuit

$$f(x) = \mathrm{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Main Steps

1. Set as many **levels** as **distinct** powers of two in the WTF.
2. Use **threshold gates** at each level to **add** the variables and carry bits from previous level.
3. The result acts as the carry bits for the next level by **appropriate wiring**.
4. **Accept** if the carry from the top level is **1**.



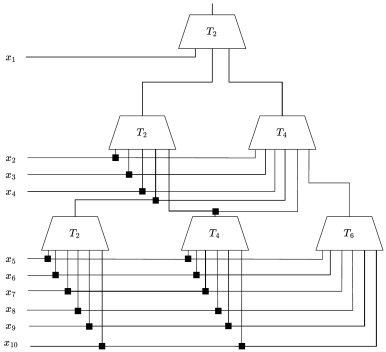
Theorem

This circuit has $2n^2 \log \sigma$ wires.

Revisiting Beimel & Weinreb Schemes: Optimization

$$f(x) = \mathrm{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Idea: To **reduce** the number of **used wires** by adding **auxiliary OR** gates.



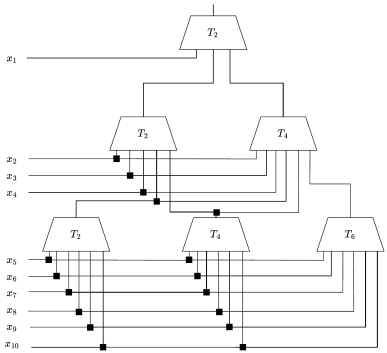
Revisiting Beimel & Weinreb Schemes: Optimization

$$f(x) = \text{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Idea: To **reduce** the number of **used wires** by adding **auxiliary OR** gates.

Main Steps

1. Add **OR gates** to track only the **highest threshold gate** activated at each level.



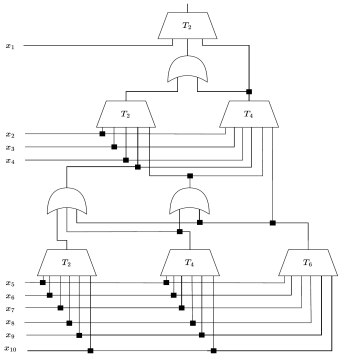
Revisiting Beimel & Weinreb Schemes: Optimization

$$f(x) = \text{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Idea: To **reduce** the number of **used wires** by adding **auxiliary OR** gates.

Main Steps

1. Add **OR gates** to track only the **highest threshold gate** activated at each level.



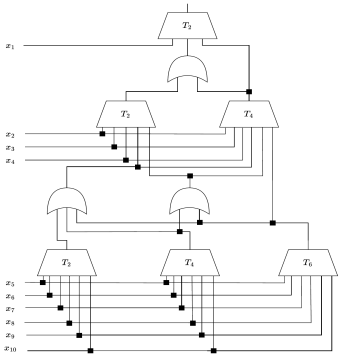
Revisiting Beimel & Weinreb Schemes: Optimization

$$f(x) = \text{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Idea: To **reduce** the number of **used wires** by adding **auxiliary OR** gates.

Main Steps

1. Add **OR gates** to track only the **highest threshold gate** activated at each level.
2. The rest of the circuit **remains the same**.



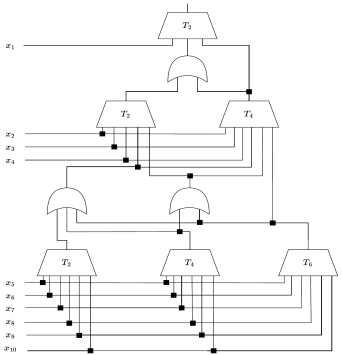
Revisiting Beimel & Weinreb Schemes: Optimization

$$f(x) = \text{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Idea: To **reduce** the number of **used wires** by adding **auxiliary OR** gates.

Main Steps

1. Add **OR gates** to track only the **highest threshold gate** activated at each level.
2. The rest of the circuit **remains the same**.
3. For any input, it suffices to use the wires of the **highest threshold gate** activated at each level.



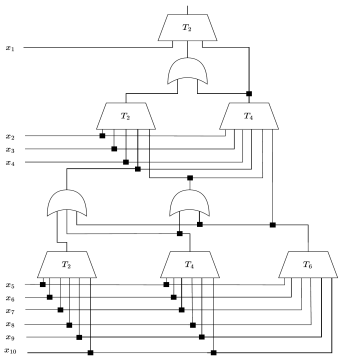
Revisiting Beimel & Weinreb Schemes: Optimization

$$f(x) = \text{sign}(4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} - 8)$$

Idea: To **reduce** the number of **used wires** by adding **auxiliary OR** gates.

Main Steps

1. Add **OR gates** to track only the **highest threshold gate** activated at each level.
2. The rest of the circuit **remains the same**.
3. For any input, it suffices to use the wires of the **highest threshold gate** activated at each level.



Theorem

Computing the output from any given input requires only $4n \log \sigma$ wires.

Revisiting Beimel & Weinreb Schemes: Our Results

Theorem

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\boldsymbol{4\lambda n^2 \log \sigma}$
- usage of only $\boldsymbol{4\lambda n \log \sigma}$ bits of the public information during reconstruction

Revisiting Beimel & Weinreb Schemes: Our Results

Theorem

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

- share size $\boldsymbol{\lambda}$
- public information size $\boldsymbol{4\lambda n^2 \log \sigma}$
- usage of only $\boldsymbol{4\lambda n \log \sigma}$ bits of the public information during reconstruction

Proof: Apply **Yao's compiler** to our circuit.

Revisiting Beimel & Weinreb Schemes: Our Results

Theorem

Given a security parameter $\lambda \in \mathbb{N}$, WTASs admit **computational** schemes with

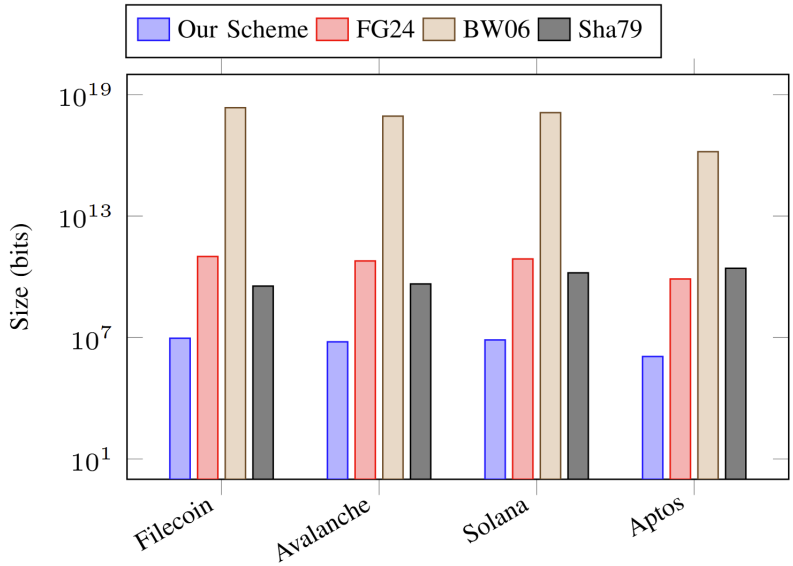
- share size $\boldsymbol{\lambda}$
- public information size $\boldsymbol{4\lambda n^2 \log \sigma}$
- usage of only $\boldsymbol{4\lambda n \log \sigma}$ bits of the public information during reconstruction

Proof: Apply **Yao's compiler** to our circuit.

Information-Theoretic Scheme

Applying balancing and dynamic programming to our circuit yields an equivalent information-theoretic scheme to that of Beimel and Weinreb.

Revisiting Beimel & Weinreb Schemes: Comparison



Closing Remarks & Open Questions

Closing Remarks

- The **share size** of existing schemes for WTASs have a **strong dependency** on the **size of the weights**.
- The most efficient **information-theoretic** scheme has **superpolynomial** share size.
- Moving towards the **computational** setting allows more efficient schemes.
- Our **computational** scheme provides a **concrete** and **efficient** share size.

Closing Remarks & Open Questions

Closing Remarks

- The **share size** of existing schemes for WTASs have a **strong dependency** on the **size of the weights**.
- The most efficient **information-theoretic** scheme has **superpolynomial** share size.
- Moving towards the **computational** setting allows more efficient schemes.
- Our **computational** scheme provides a **concrete** and **efficient** share size.

Open Questions

- Is there any scheme with **polynomial** share size for WTASs?
- Can we improve the $\boldsymbol{\Omega(\sqrt{n})}$ **lower bound** for these access structures?
- Can we make our scheme **linear**?
- Are weighted threshold functions in $\mathbf{mNC^1}$?

Moltes Gràcies!

Check the Full Version of our Paper!