Timed-Release
Secret Sharing Schemes with Information Theoretic Security

Yohei Watanabe and Junji Shikata
Yokohama National University, Japan
Secret Sharing Scheme and Timed-Release Functionality

- Secret sharing (SS) scheme [Sha79,Bla79] is an important primitive.
- Cryptographic functionality associated with “time” is useful.
  - Concept of “time” is inseparable from our lives.
  - Such an well-known functionality is: Timed-Release Functionality.
Secret Sharing Scheme and Timed-Release Functionality

◆ Secret sharing (SS) scheme [Sha79, Bla79] is an important primitive.

◆ Cryptographic functionality associated with “time” is useful.
  ◆ Concept of “time” is inseparable from our lives.
  ◆ Such an well-known functionality is: Timed-Release Functionality.

“Can we realize a secret sharing scheme with timed-release functionality?”

◆ We focus on Timed-Release Secret Sharing Schemes.
Secret Sharing Scheme and Timed-Release Functionality

◆ Secret sharing (SS) scheme [Sha79, Bla79] is an important primitive.

◆ Cryptographic functionality associated with “time” is useful.
  ◆ Concept of “time” is inseparable from our lives.
  ◆ Such an well-known functionality is: Timed-Release Functionality.

“Can we realize a secret sharing scheme with timed-release functionality?”

◆ We focus on Timed-Release Secret Sharing Schemes.

Related Works

➤ Timed-Release Computational Secret Sharing Scheme [WS14]
  ➤ Presented at ProvSec 2014 last week.
**Security**

**Computational Security**
- Underlying main theory: **Complexity theory**.
- Based on **computational assumption**.
- The adversary has **polynomial-time computational power**.

**Unconditional Security**
(Information-Theoretic Security)
- Underlying main theories: **Information theory** and **Probability theory**.
- Based on some assumption, but **no computational assumption is required**.
- The adversary has **infinite computational power**.
Security

Computational Security
- Underlying main theory: Complexity theory.
- Based on computational assumption.
- The adversary has polynomial-time computational power.

Unconditional Security (Information-Theoretic Security)
- Underlying main theories: Information theory and Probability theory.
- Based on some assumption, but no computational assumption is required.
- The adversary has infinite computational power.

Development of Algorithms

Realization of Quantum Computer
Security

Computational Security

- Underlying main theory: Complexity theory.
- Based on computational assumption.
- The adversary has polynomial-time computational power.

Unconditional Security (Information-Theoretic Security)

- Underlying main theories: Information theory and Probability theory.
- Based on some assumption, but no computational assumption is required.
- The adversary has infinite computational power.

Development of Algorithms

Realization of Quantum Computer

The possibility that some computational assumptions are broken.
Shannon Entropy

- Shannon entropy $H(\cdot)$
  - Measure of the uncertainty of random variable.

\[
H(X) := -\sum_{x \in \mathcal{X}} \Pr(X = x) \log \Pr(X = x),
\]

where $X$ is a random variable which takes a value on a set $\mathcal{X}$. 
Shannon Entropy

◆ **Shannon entropy** $H(\cdot)$

- Measure of the uncertainty of random variable.

\[
H(X) := - \sum_{x \in \mathcal{X}} \Pr(X = x) \log \Pr(X = x),
\]

where $X$ is a random variable which takes a value on a set $\mathcal{X}$.

◆ **Conditional Entropy** $H(\cdot | \cdot)$.

\[
H(X | Y) := \sum_{y \in \mathcal{Y}} \Pr(Y = y) H(X | Y = y).
\]
\[(k,n)\text{-threshold Secret Sharing } ((k,n)\text{-SS})\]

\[P := \{P_1, P_2, ..., P_n\}.\]
(k,n)-threshold Secret Sharing ((k,n)-SS)

\[ P := \{P_1, P_2, \ldots, P_n\} \]

secret \( S \)

dealer \( D \)

\( n \) shares

share \( u_1 \) to participant \( P_1 \)

share \( u_k \) to participant \( P_k \)

share \( u_n \) to participant \( P_n \)
(k,n)-threshold Secret Sharing ((k,n)-SS)

\[ \mathcal{P} := \{P_1, P_2, \ldots, P_n\}. \]

Secret can be reconstructed from at least k shares.
(k,n)-threshold Secret Sharing ((k,n)-SS)

\[ P := \{P_1, P_2, \ldots, P_n\}. \]

\[ H(S \mid U_A) = 0 \]
\[ (A \subset P, k \leq |A| \leq n). \]

Secret can be reconstructed from at least k shares
(k,n)-threshold Secret Sharing ((k,n)-SS)

\( P := \{P_1, P_2, ..., P_n\} \).

\( H(S \mid U_A) = 0 \quad (A \subset P, k \leq |A| \leq n). \)

Secret can be reconstructed from at least k shares.

No information is leaked from at most k-1 shares.

Secret S

dealer D

\( \mathcal{P} \)

n shares

participant \( P_1 \)

share \( u_1 \)

participant \( P_k \)

share \( u_k \)

participant \( P_n \)

share \( u_n \)
(k,n)-threshold Secret Sharing ((k,n)-SS)

\[ P := \{P_1, P_2, \ldots, P_n\}. \]

No information is leaked from at most \(k-1\) shares

Secret can be reconstructed from at least \(k\) shares

\[ H(S \mid U_A) = 0 \]
\[ (A \subset P, k \leq |A| \leq n). \]

\[ H(S \mid U_F) = H(S) \]
\[ (F \subset P, 1 \leq |F| \leq k - 1). \]
Timed-Release Cryptography

Goal: securely send certain information into the future.

Example: Timed-Release Public-Key Encryption (TR-PKE) [RSW96]

\[ t_0 \]

Timed-Release Cryptography

Goal: securely send certain information into the future.

Example: Timed-Release Public-Key Encryption (TR-PKE) [RSW96]

**Timed-Release Cryptography**

Goal: securely send certain information into the future.

**Example: Timed-Release Public-Key Encryption (TR-PKE)**

Timed-Release Cryptography

Goal: securely send certain information into the future.

Example: Timed-Release Public-Key Encryption (TR-PKE) [RSW96]

Timed-Release Cryptography

Goal: securely send certain information into the future.

Example: Timed-Release Public-Key Encryption (TR-PKE) [RSW96]

Our Proposal

Two kinds of Timed-Release Secret Sharing (TR-SS) Schemes

- \((k, n)\)-TR-SS: Realize reconstruction with timed-release functionality.
  - Formalize a model and security notions.
  - Derive lower bounds on sizes of shares, time-signals and secret keys.
  - Propose an optimal direct construction in the sense that it meets equality in the above every bound.

- \((k_1, k_2, n)\)-TR-SS: Realize timed-release functionality and secret sharing functionality \textit{simultaneously}.
  - Formalize a model and security notions.
  - Derive lower bounds on sizes of shares, time-signals and secret keys.
  - Show a naïve construction is not optimal.
  - Propose an optimal direct (but restricted) construction.
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)

dealer $D$

secret $s$

participant $P_1$

participant $P_k$

participant $P_n$

$T$ : specified time
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)

dealer D

secret s

T : specified time

share $u_1^{(T)}$

share $u_k^{(T)}$

share $u_n^{(T)}$

participant $P_1$

participant $P_k$

participant $P_n$

time-server TS
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)

No secret can be reconstructed without the time-signal at the specified time.
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)

No secret can be reconstructed without the time-signal at the specified time

T : specified time

dealer D

secret s

share $u^{(T)}_1$

participant $P_1$

share $u^{(T)}_k$

participant $P_k$

share $u^{(T)}_n$

participant $P_n$

time-signal at time $T$

$t_{s(T)}$

time-server TS

No secret can be reconstructed without the time-signal at the specified time
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)

No secret can be reconstructed without the time-signal at the specified time.

Secret can be reconstructed from at least k shares and the time-signal at the specified time.
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)

**No** secret can be reconstructed without the time-signal at the specified time.

Secret can be reconstructed from at least k shares and the time-signal at the specified time.

\[ H \left( S \mid U_A^{(T)}, T \right) = 0 \quad (A \subseteq P, k \leq |A| \leq n). \]
(k,n)-Timed-Release Secret Sharing
((k,n)-TR-SS)

No secret can be reconstructed without the time-signal at the specified time.

Share $u_1^{(T)}$ to participant $P_1$

Share $u_k^{(T)}$ to participant $P_k$

Share $u_n^{(T)}$ to participant $P_n$

Secret can be reconstructed from at least $k$ shares and the time-signal at the specified time.

No information is leaked from at most $k - 1$ shares.
(k,n)-TR-SS: Model

**Entities.**
A dealer \( \mathbf{D} \), \( n \) participants \( \mathcal{P} := \{ P_1, \ldots, P_n \} \), a time-server \( \mathbf{TS} \), and a trusted authority \( \mathbf{TA} \).

**Phases.**
*Initialize, Share, Extract* and *Reconstruct.*

**Spaces.**
- \( S \) : a set of secrets;
- \( SK \) : a set of secret keys;
- \( T := \{1,2,\ldots,\tau\} \) : a set of time;
- \( \mathcal{U} \) : a set of shares, where \( \mathcal{U} := \bigcup_{i=1}^{n} U_i \) and \( U_i := \bigcup_{t=1}^{\tau} U_i^{(t)} \);
- \( \mathcal{T} \) : a set of time-signals, where \( \mathcal{T} := \bigcup_{t=1}^{\tau} \mathcal{T}^{(t)} \).
(k,n)-TR-SS: Model

1. **Initialize.**
   1. **TA** generates a secret key $sk \in SK$ for **TS** and **D**.
   2. **TA** distributes $sk$ to **TS** and **D** via secure channels.
   3. **TA** deletes $sk$ from his memory.
(k,n)-TR-SS: Model

2. **Share.**
   1. D randomly selects a secret $s \in S$ and chooses $k$ and $n$.
   2. D specifies future time $T \in \mathcal{T}$, and computes $n$ shares $u_1^{(T)}, \ldots, u_n^{(T)}$.
   3. D sends $(u_i^{(T)}, T)$ to $P_i$ via a secure channel ($i = 1, 2, \ldots, n$).
(k,n)-TR-SS: Model

3. Extract.

1. At each time \( t \in \mathcal{T} \), TS generates a time-signal \( ts^{(t)} \) by using his secret key \( sk \).
2. TS broadcasts \( ts^{(t)} \).

For simplicity, we assume \( ts^{(t)} \) is deterministically computed by \( t \) and \( sk \).
(k,n)-TR-SS: Model

4. **Reconstruct.**

At the specified time $T$, any set of participants $A := \{P_{i_1}, ..., P_{i_j}\}$ ($k \leq j \leq n$) can reconstruct $s$ from their shares $u_{i_1}^{(T)}, ..., u_{i_j}^{(T)}$ and a time-signal $t_s^{(T)}$ at the specified time $T$. 
(k,n)-TR-SS: Security

We consider two kinds of security.

(i) Traditional secret sharing security.

(ii) Timed-release security.

Formally, a (k,n)-TR-SS scheme is secure if the following conditions are satisfied.

(i) For any $F \subset P$ s.t. $1 \leq |F| \leq k - 1$ and any $T \in \mathcal{T}$, it holds that

$$H \left( S \mid U_F^{(T)}, TI^{(1)}, ..., TI^{(\tau)} \right) = H(S).$$

(ii) For any $A \subset P$ s.t. $k \leq |A| \leq n$ and any $T \in \mathcal{T}$, it holds that

$$H \left( S \mid U_A^{(T)}, TI^{(1)}, ..., TI^{(T-1)}, TI^{(T+1)}, ..., TI^{(\tau)} \right) = H(S).$$
(k,n)-TR-SS: Tight Lower Bounds

Lower bounds on sizes of shares, time-signals and secret keys required for a secure (k,n)-TR-SS scheme as follows.

**Theorem.**
For any $i \in \{1, 2, \ldots, n\}$ and for any $T \in \mathcal{T}$, we have

(i) $H\left(U_i^{(T)}\right) \geq H(S)$,

(ii) $H(TI^{(T)}) \geq H(S)$,

(iii) $H(SK) \geq \tau H(S)$.

A construction of a secure (k,n)-TR-SS scheme is said to be **optimal** if it meets equality in every bound of (i)-(iii) in the above theorem.
(k,n)-TR-SS: Tight Lower Bounds

Lower bounds on sizes of shares, time-signals and secret keys required for a secure (k,n)-TR-SS scheme as follows.

Theorem.

For any \(i \in \{1, 2, ..., n\}\) and for any \(T \in \mathcal{T}\), we have

\[
\begin{align*}
(i) & \quad H\left(U^T_i\right) \geq H(S), \\
(ii) & \quad H(TI^T) \geq H(S), \\
(iii) & \quad H(SK) \geq \tau H(S).
\end{align*}
\]

Timed-release property can be realized without any additional redundancy in the share size.

A construction of a secure (k,n)-TR-SS scheme is said to be optimal if it meets equality in every bound of (i)-(iii) in the above theorem.
(k, n)-TR-SS: Optimal Construction

1. **Initialize.**
   
   Let \( q \) be a prime power, where \( q > \max(n, \tau) \).
   Let \( F_q \) be a finite field with \( q \) elements.

   1. **TA** chooses \( \tau \) numbers \( r^{(j)} (j = 1, \ldots, \tau) \) from \( F_q \) uniformly at random.
   2. **TA** sends \( sk := (r^{(1)}, \ldots, r^{(\tau)}) \) to **TS** and **D**, respectively.
2. Share.

1. D randomly selects a secret $s \in \mathbb{F}_q$ and chooses $k$ and $n$.

2. D specifies future time $T \in \mathcal{T}$.

3. D randomly chooses $f(x) := c^{(T)} + \sum_{i=1}^{k-1} a_i x^i$ over $\mathbb{F}_q$, where $c^{(T)} := s + r^{(T)}$ and each $a_i$ is chosen from $\mathbb{F}_q$ uniformly at random.

4. D computes $u_i^{(T)} := f(P_i)$ and sends $(u_i^{(T)}, T)$ to $P_i$ via a secure channel ($i = 1, 2, \ldots, n$).
3. **Extract.**

At each time $t \in \mathcal{T}$, TS broadcasts $t$-th key $r^{(t)}$ as a time-signal at time $t$. 
(k,n)-TR-SS: Optimal Construction

4. **Reconstruct.**

1. A set of at least $k$ participants $A := \{P_{i_1},...,P_{i_j}\}$ can compute $c^{(T)}$ by Lagrange interpolation from their $k$ shares:

   $$c^{(T)} = \sum_{j=1}^{k} \left( \prod_{l \neq j} \frac{P_{i_j}}{P_{i_j} - P_{i_l}} \right) f(P_{i_j}).$$

2. After receiving $ts^{(T)} = r^{(T)}$, they can compute $s = c^{(T)} - r^{(T)}$. 
(k,n)-TR-SS: Optimal Construction

4. Reconstruct.

1. A set of at least $k$ participants $A := \{P_{i_1}, ..., P_{i_j}\}$ can compute $c^{(T)}$ by Lagrange interpolation from their $k$ shares:

$$c^{(T)} = \sum_{j=1}^{k} \left( \prod_{l \neq j} \frac{P_{i_j}}{P_{i_j} - P_{i_l}} \right) f (P_{i_j}).$$

2. After receiving $ts^{(T)} = r^{(T)}$, they can compute $s = c^{(T)} - r^{(T)}$.

**Theorem.**

The resulting (k,n)-TR-SS scheme by this construction is secure and optimal.
$(k_1,k_2,n)$-TR-SS

$T : \text{specified time}$

$D$  

secret $s$
$(k_1, k_2, n)$-TR-SS

$T : \text{specified time}$

$D$ secret $s$ share $u_1^{(T)}$ share $u_{k_1}^{(T)}$ share $u_{k_2}^{(T)}$ share $u_n^{(T)}$

$P_1$ $P_{k_1}$ $P_{k_2}$ $P_n$
At least $k_1$ (but no more than $k_2$) participants cannot reconstruct the secret without the time-signal at the specified time.

\((k_1, k_2, n)\)-TR-SS
(k₁,k₂,n)-TR-SS

At least k₁ (but no more than k₂) participants cannot reconstruct the secret without the time-signal at the specified time.
At least $k_1$ (but no more than $k_2$) participants cannot reconstruct the secret without the time-signal at the specified time.

Secret can be reconstructed from at least $k_1$ shares and the time-signal at the specified time.

$$H \left( S \mid U_{A}^{(T)}, TI^{(T)} \right) = 0 \quad (A \subset P, k_1 \leq |A| < k_2).$$
At least $k_1$ (but no more than $k_2$) participants cannot reconstruct the secret without the time-signal at the specified time.

Secret can be reconstructed from at least $k_1$ shares and the time-signal at the specified time.

No information is leaked from at most $k_1 - 1$ shares.
The diagram illustrates a $(k_1, k_2, n)$-TR-SS scheme. The secret can be reconstructed from only at least $k_2$ shares. The equation is:

$$H \left( S \mid U^{(T)}_{\hat{A}} \right) = 0$$

where $(\hat{A} \subset P, k_2 \leq |\hat{A}| \leq n)$. The diagram shows the distribution of shares $u_1^{(T)}$, $u_{k_1}^{(T)}$, $u_{k_2}^{(T)}$, and $u_n^{(T)}$ to $P_1$, $P_{k_1}$, $P_{k_2}$, and $P_n$ respectively, along with the time specification $T$. The secret can be reconstructed only when $k_2$ shares are available.
(k_1,k_2,n)-TR-SS: Model

Entities.
A dealer D, n participants \( \mathcal{P} := \{P_1, \ldots, P_n\} \), a time-server TS, and a trusted authority TA.

Phases.
Initialize, Share, Extract, Reconstruct with time-signals, and Reconstruct without time-signals.

Spaces.
\( S \): a set of secrets;
\( S\mathcal{K} \): a set of secret keys;
\( \mathcal{T} := \{1, 2, \ldots, \tau\} \): a set of time;
\( \mathcal{U} \): a set of shares, where \( \mathcal{U} := \bigcup_{i=1}^{n} \mathcal{U}_i \) and \( \mathcal{U}_i := \bigcup_{t=1}^{\tau} \mathcal{U}_i^{(t)} \);
\( \mathcal{T}\mathcal{I} \): a set of time-signals, where \( \mathcal{T}\mathcal{I} := \bigcup_{t=1}^{\tau} \mathcal{T}\mathcal{I}^{(t)} \).
(k₁,k₂,n)-TR-SS: Model

1. **Initialize.** (the same procedure as that in (k,n)-TR-SS)
   1. **TA** generates a secret key \( sk \in SK \) for **TS** and **D**.
   2. **TA** distributes \( sk \) to **TS** and **D** via secure channels.
   3. **TA** deletes \( sk \) from his memory.

![Diagram showing TA generating and distributing a secret key to TS and D via secure channels.](Secure Channel)
(k₁, k₂, n)-TR-SS: Model

2. **Share.**

1. D randomly selects a secret $s \in S$ and chooses $k₁, k₂$ and $n$.
2. D specifies future time $T \in \mathcal{T}$, and computes $n$ shares $u₁(T), ..., u_n(T)$.
3. D sends $(u_i(T), T)$ to $P_i$ via a secure channel ($i = 1, 2, ..., n$).
(\(k_1, k_2, n\))-TR-SS: Model

3. **Extract**. (the same procedure as that in \((k,n)\)-TR-SS)

1. At each time \(t \in \mathcal{T}\), TS generates a time-signal \(ts(t) \in \mathcal{T}\mathcal{T}(t)\) by using his secret key \(sk\).

2. TS broadcasts \(ts(t)\).

For simplicity, we assume \(ts(t)\) is deterministically computed by \(t\) and \(sk\).
(k₁,k₂,n)-TR-SS: Model

4. **Reconstruct with time-signals.**

At the specified time $T$, any set of participants $A := \{P_{i_1}, ..., P_{i_j}\}$ ($k_1 \leq j < k_2$) can reconstruct $s$ from their shares $u_{i_1}^{(T)}, ..., u_{i_j}^{(T)}$ and a time-signal $t_{s}^{(T)}$ at the specified time $T$. 

![Diagram](image)
(\(k_1, k_2, n\))-TR-SS: Model

5. **Reconstruct without time-signals.**

At anytime, any set of participants \(A \coloneqq \{P_{i_1}, \ldots, P_{i_j}\} (k_2 \leq j \leq n)\) can reconstruct \(s\) from **only** their shares \(u_{i_1}^{(T)}, \ldots, u_{i_j}^{(T)}\).
(k₁,k₂,n)-TR-SS: Security

We consider two kinds of security.

(i) Traditional secret sharing security.

(ii) Timed-release security.

Formally, a (k₁,k₂,n)-TR-SS scheme is secure if the following conditions are satisfied.

(i) For any F ⊂ P s.t. 1 ≤ |F| ≤ k₁ − 1 and any T ∈ 𝒯, it holds that

\[ H \left( S \mid U_F^{(T)}, TI^{(1)}, \ldots, TI^{(τ)} \right) = H(S). \]

(ii) For any 𝔻 ⊂ P s.t. k₁ ≤ |𝔻| < k₂ and any T ∈ 𝒯, it holds that

\[ H \left( S \mid U_𝔻^{(T)}, TI^{(1)}, \ldots, TI^{(T−1)}, TI^{(T+1)}, \ldots, TI^{(τ)} \right) = H(S). \]
(\(k_1, k_2, n\))-TR-SS: Tight Lower Bounds

Lower bounds on sizes of shares, time-signals and secret keys required for a secure \((k_1, k_2, n)\)-TR-SS scheme as follows.

**Theorem.**

For any \(i \in \{1, 2, \ldots, n\}\) and for any \(T \in \mathcal{T}\), we have

\[
(i) \quad H(U_i^{(T)}) \geq H(S).
\]

If (i) holds with equality (i.e. \(H(U_i^{(T)}) = H(S)\) for any \(i\) and \(T\)), we have

\[
(ii) \quad H(TI^{(T)}) \geq (k_2 - k_1)H(S),
\]

\[
(iii) \quad H(SK) \geq \tau(k_2 - k_1)H(S).
\]

A construction of a secure \((k_1, k_2, n)\)-TR-SS scheme is said to be **optimal** if it meets equality in every bound of (i)-(iii) in the above theorem.
(\(k_1, k_2, n\))-TR-SS: Naïve Construction

We can realize a secure \((k_1, k_2, n)\)-TR-SS scheme by combining the following two schemes.

- A secure \((k_1, n)\)-TR-SS scheme (the first scheme)
- A secure \((k_2, n)\)-SS scheme (e.g. Shamir’s scheme)

However, the resulting scheme is NOT optimal.

- The share size is **twice** as large as the underlying secret size.
(k_1, k_2, n)-TR-SS: Constructing Idea

To achieve an optimal construction, we use the technique in [JS13]:

In the phase **Share**,

- D computes **public parameters**, and
- the public parameters are broadcasted to participants,
- or else stored on a publicly accessible authenticated bulletin board.
(k₁,k₂,n)-TR-SS: Optimal Construction

1. **Initialize.**

   Let \( q \) be a prime power, where \( q > \max(n, \tau) \).
   Let \( \mathbb{F}_q \) be a finite field with \( q \) elements.

   1. **TA** chooses \( \ell \), which is the maximum difference between \( k_2 \) and \( k_1 \).
   2. **TA** chooses \( \ell \cdot \tau \) numbers \( r^{(t)}_i \) (\( i = 1, ..., \ell \), \( t = 1, ..., \tau \)) from \( \mathbb{F}_q \) uniformly at random.
   3. **TA** sends \( sk := \left\{ (r^{(t)}_1, ..., r^{(t)}_\ell) \right\}_{1 \leq t \leq \tau} \) to **TS** and **D**, respectively.
(k₁,k₂,n)-TR-SS: Optimal Construction

1. Initialize.

Let \( q \) be a prime power, where \( q > \max(n, \tau) \).
Let \( \mathbb{F}_q \) be a finite field with \( q \) elements.

1. \( TA \) chooses \( \ell \), which is the maximum difference between \( k₂ \) and \( k₁ \).
2. \( TA \) chooses \( \ell \cdot \tau \) numbers \( r_{i}^{(t)} \) (\( i = 1, \ldots, \ell, \ t = 1, \ldots, \tau \)) from \( \mathbb{F}_q \) uniformly at random.
3. \( TA \) sends \( sk := \left\{ (r_{1}^{(t)}, \ldots, r_{\ell}^{(t)}) \right\}_{1 \leq t \leq \tau} \) to \( TS \) and \( D \), respectively.

Note.
This construction is optimal but restricted, since \( D \) will be only allowed to choose \( k₁ \) and \( k₂ \) s.t. \( k₂ - k₁ \leq \ell \) in the phase Share.
(k_1,k_2,n)-TR-SS: Optimal Construction

2. Share.

1. D randomly selects a secret \( s \in \mathbb{F}_q \) and chooses \( k_1, k_2 \) and \( n \).
2. D specifies future time \( T \in \mathcal{T} \).
3. D randomly chooses
   \[
   f(x) := s + a_1 x + \ldots + a_{k_1-1} x^{k_1-1} + a_{k_1} x^{k_1} + \ldots + a_{k_2-1} x^{k_2-1},
   \]
   over \( \mathbb{F}_q \), where each \( a_i \) is chosen from \( \mathbb{F}_q \) uniformly at random.
4. D computes \( u_i^{(T)} := f(P_i) \) and \( p_i^{(T)} := a_{k_1-1+i} + r_i^{(T)} \) \((i = 1, \ldots, k_2 - k_1)\).
5. D sends \( (u_i^{(T)}, T) \) to \( P_i \) via a secure channel \((i = 1,2,\ldots,n)\) and disclose \( p_1^{(T)}, \ldots, p_{k_2-k_1}^{(T)} \).
(\(k_1, k_2, n\))-TR-SS: Optimal Construction

2. **Share.**

1. D randomly selects a secret \(s \in \mathbb{F}_q\) and chooses \(k_1, k_2\) and \(n\).
2. D specifies future time \(T \in \mathcal{T}\).
3. D randomly chooses

   \[
   f(x) := s + a_1 x + \cdots + a_{k_1 - 1} x^{k_1 - 1} + a_{k_1} x^{k_1} + \cdots + a_{k_2 - 1} x^{k_2 - 1},
   \]

   over \(\mathbb{F}_q\), where each \(a_i\) is chosen from \(\mathbb{F}_q\) uniformly at random.
4. D computes \(u_i^{(T)} := f(P_i)\) and \(p_i^{(T)} := a_{k_1 - 1 + i} + r_i^{(T)}\) \((i = 1, \ldots, k_2 - k_1)\).
5. D sends \((u_i^{(T)}, T)\) to \(P_i\) via a secure channel \((i = 1, 2, \ldots, n)\) and disclose \(p_1^{(T)}, \ldots, p_{k_2 - k_1}\).
(k_1,k_2,n)-TR-SS: Optimal Construction

3. **Extract.**

At each time $t \in T$, TS broadcasts $t$-th key $(r_1^{(t)}, \ldots, r_{\ell}^{(t)})$ as a time-signal at time $t$. 
(k₁,k₂,n)-TR-SS: Optimal Construction

4. **Reconstruct with time-signals.**

Suppose that all participants receive \( ts^{(T)} = (r_1^{(T)}, \ldots, r_\ell^{(T)}) \).

Let \( A := \{P_{i_1}, \ldots, P_{i_{k_1}}\} \) be a set of any \( k_1 \) participants.

1. Each \( P_{i_j} \in A \) computes \( a_{k_1-1+k} = p_k^{(T)} - r_k^{(T)} \) (\( k = 1, \ldots, k_2 - k_1 \)) and constructs \( g(x) := a_{k_1}x^{k_1} + \cdots + a_{k_2-1}x^{k_2-1} \).

2. Each \( P_{i_j} \in A \) computes \( h(P_{i_j}) := f(P_{i_j}) - g(P_{i_j}) \) s.t.

   \[ h(x) := s + a_1 x + \cdots + a_{k_1-1} x^{k_1-1}. \]

3. \( A \) computes \( s \) by Lagrange interpolation from \( h(P_{i_1}), \ldots, h(P_{i_{k_1}}) \):

   \[
   s = \sum_{j=1}^{k_1} \left( \prod_{l \neq j} \frac{P_{i_j}}{P_{i_j} - P_{i_l}} \right) h(P_{i_j}).
   \]
(k₁,k₂,n)-TR-SS: Optimal Construction

5. **Reconstruct without time-signals.**

   1. Any set of at least \( k_2 \) participants \( \hat{A} := \{P_{i_1}, ..., P_{i_{k_2}}\} \) can compute \( s \) by Lagrange interpolation from \( f(P_{i_1}), ..., f(P_{i_{k_2}}) \):

   \[
   s = \sum_{j=1}^{k_2} \left( \prod_{l \neq j} \frac{P_{i_j}}{P_{i_j} - P_{i_l}} \right) f(P_{i_j}).
   \]
Conclusion

  ◆ One is a secret sharing scheme with timed-release functionality.
  ◆ Another one is a hybrid scheme.

◆ By using TR-SS, we can add timed-release functionality to applications of secret sharing schemes.
  ◆ Information-theoretically secure key escrow with limited time span.
  ◆ Information-theoretically secure timed-release encryption.