Timed-Release Secret Sharing Schemes with Information Theoretic Security

Yohei Watanabe and Junji Shikata

Yokohama National University, Japan

BalkanCryptSec 2014

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.

Development of Algorithms

Realization of Quantum Computer

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

The possibility that some computational assumptions are broken.

polynomial-time computational power.

Development of Algorithms

Realization of Quantum Computer

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.**

Shannon Entropy

◆ **Shannon entropy** $\boldsymbol{H}(\cdot)$

➢ Measure of the uncertainty of random variable.

$$\boldsymbol{H}(\boldsymbol{X}) \coloneqq -\sum_{\boldsymbol{x}\in\mathcal{X}} \mathbf{Pr}(\boldsymbol{X}=\boldsymbol{x}) \log \mathbf{Pr}(\boldsymbol{X}=\boldsymbol{x}),$$

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.

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

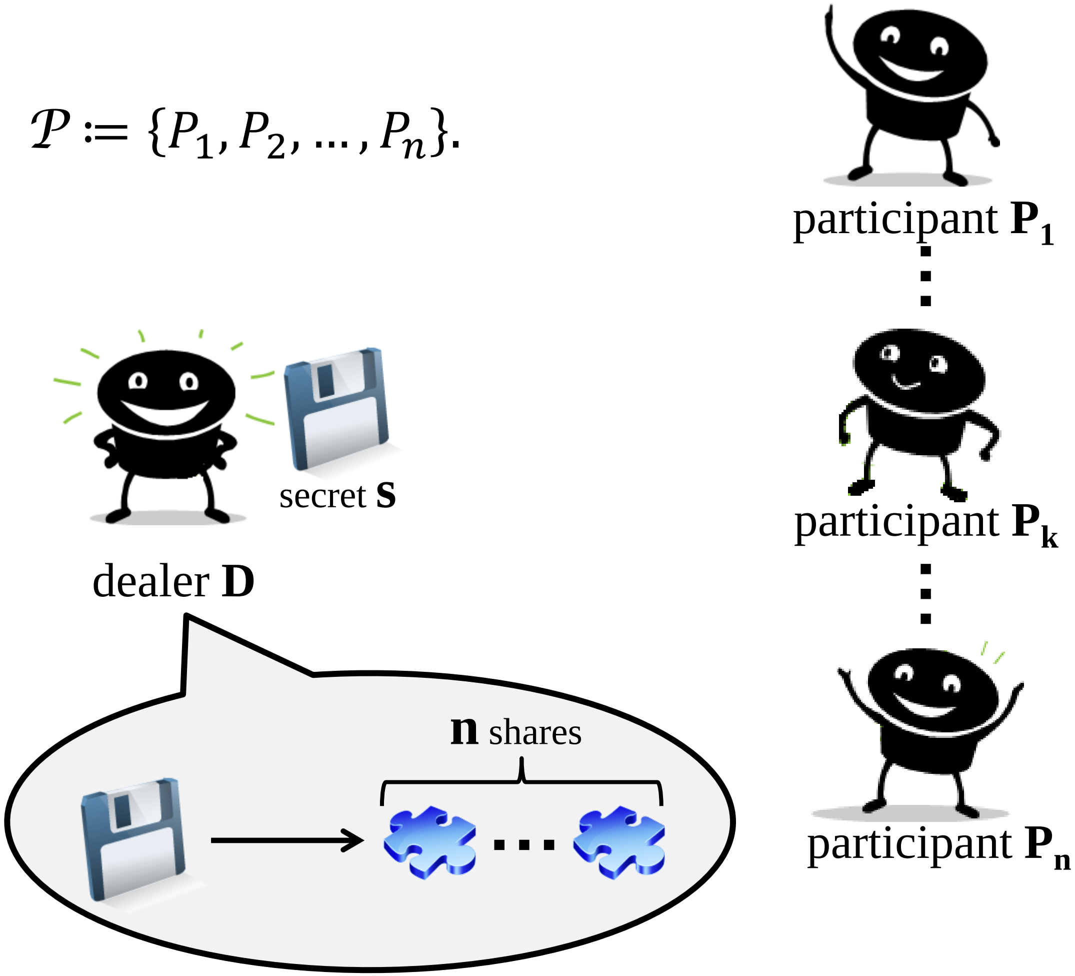
where X is a random variable which takes a value on a set $\mathcal{X}$.

- **Conditional Entropy** $H(\cdot \mid \cdot)$ **.**

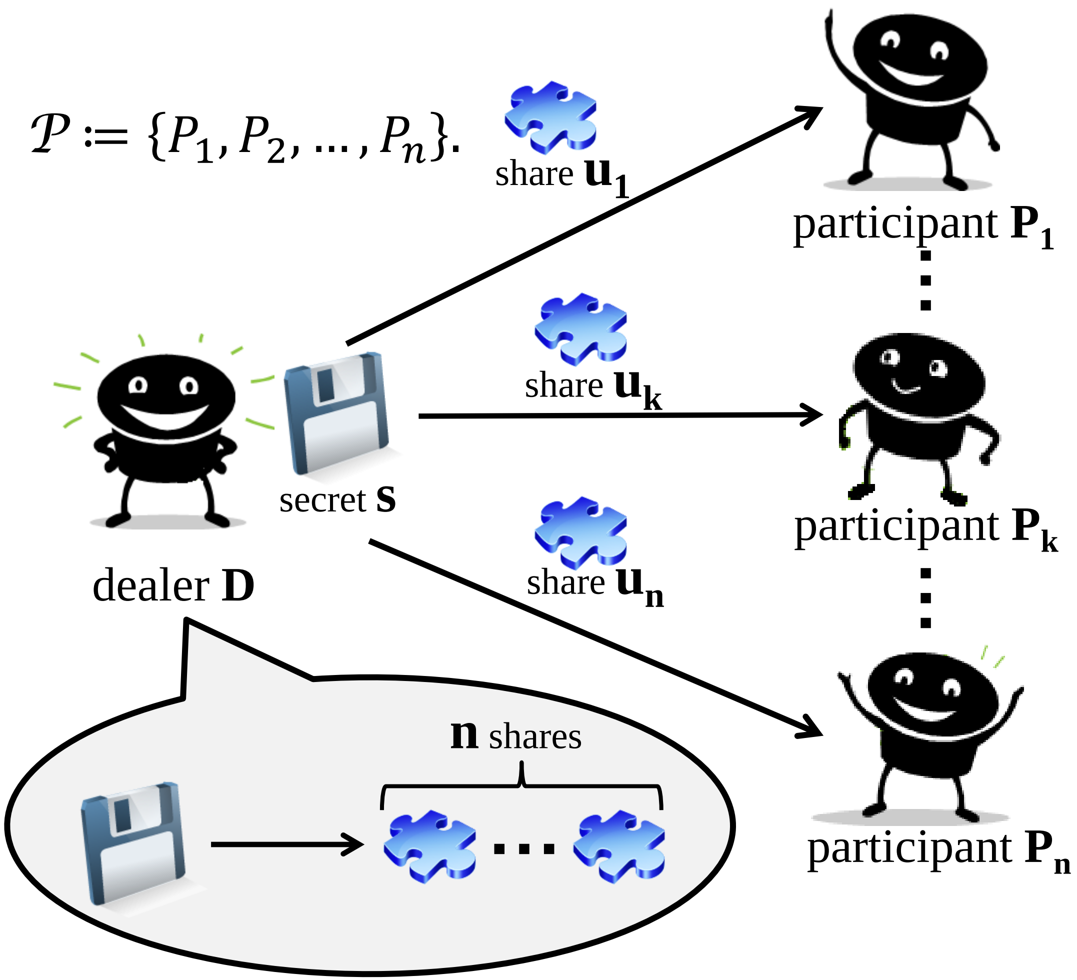
$$H(X \mid Y) := \sum_{y \in \mathcal{Y}} \mathbf{Pr}(Y = y) H(X \mid Y = y).$$

(k,n)-threshold Secret Sharing ((k,n)-SS)

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

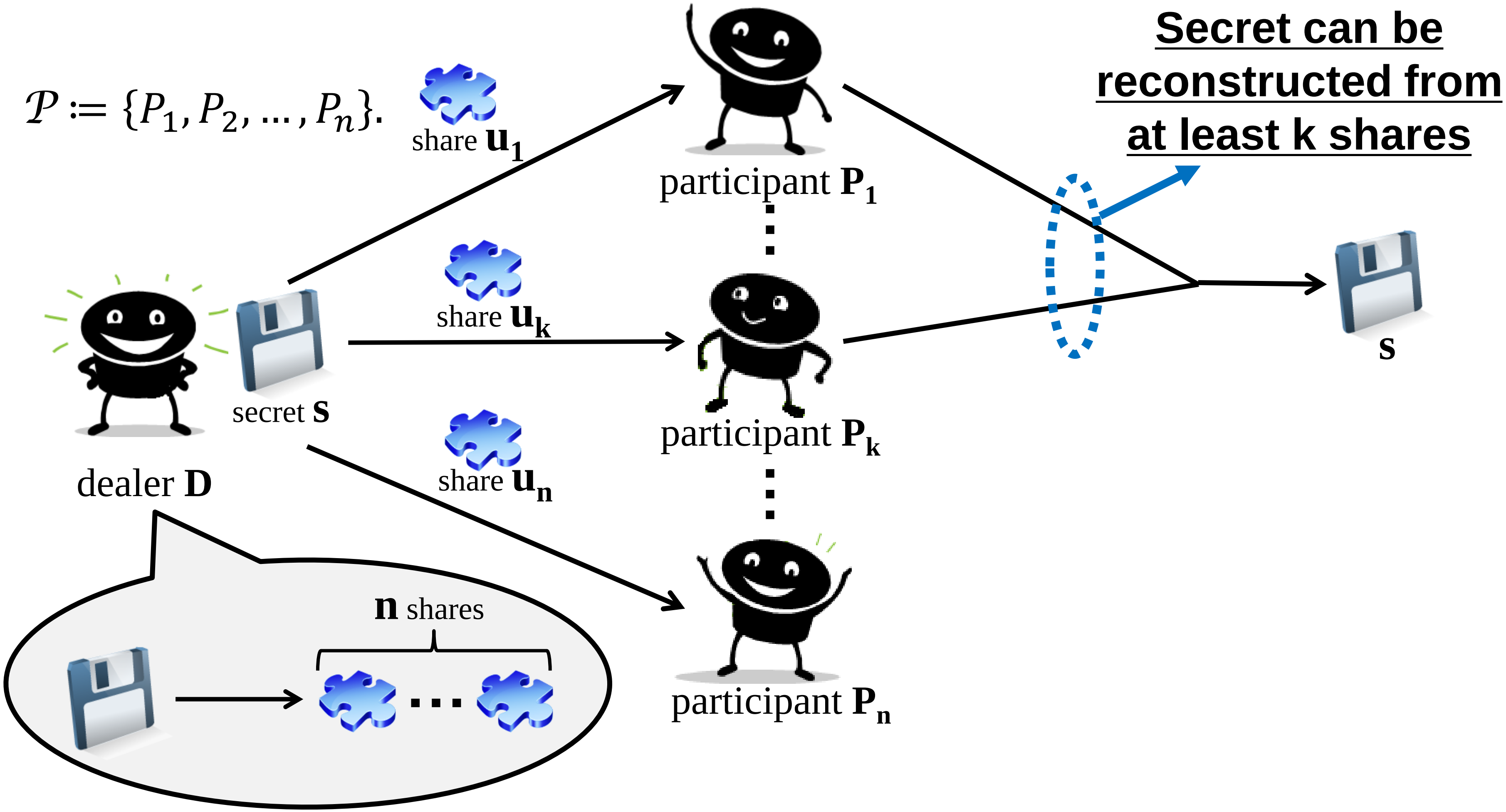


(k,n)-threshold Secret Sharing ((k,n)-SS)



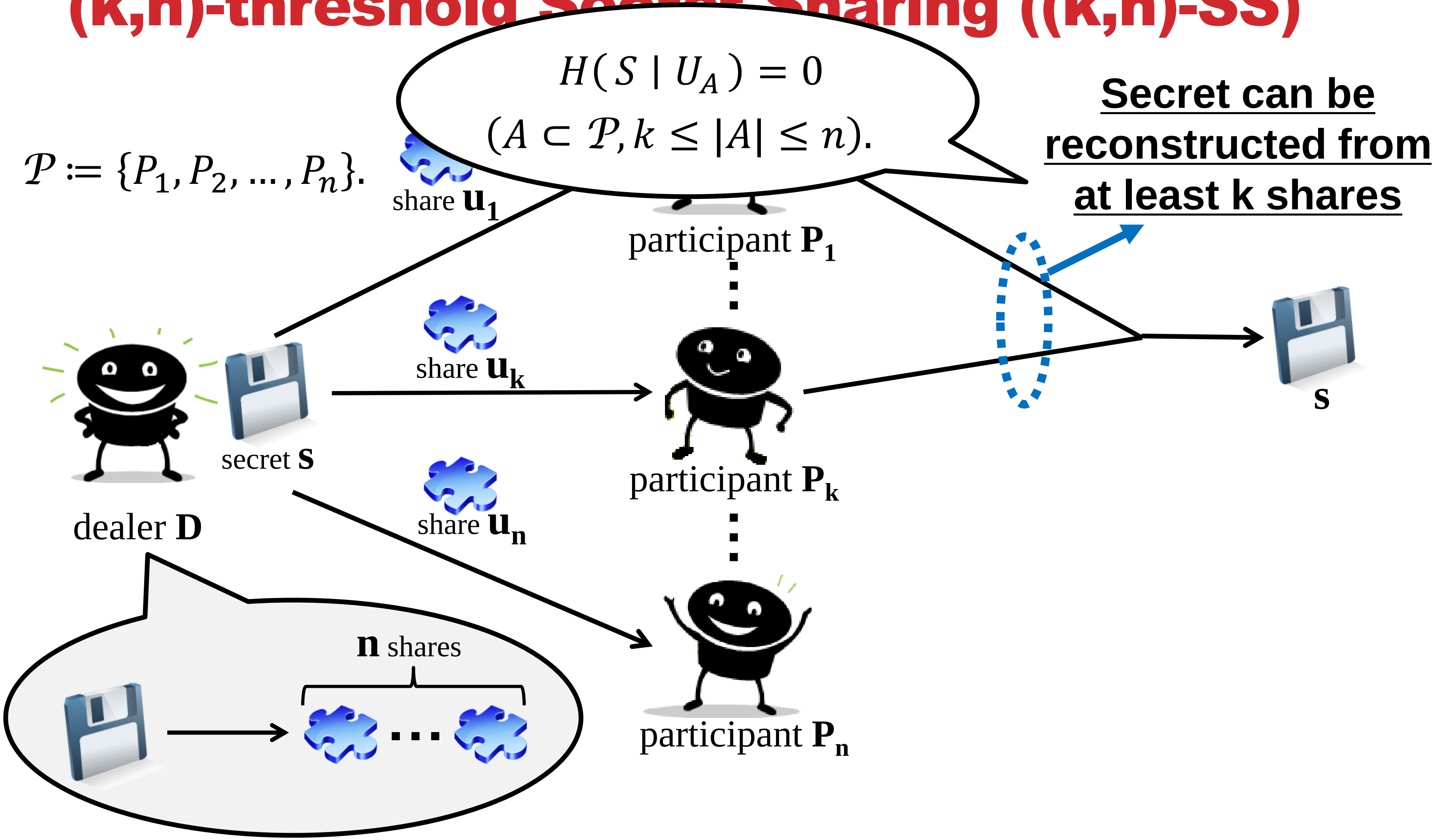
(k,n)-threshold Secret Sharing ((k,n)-SS)

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

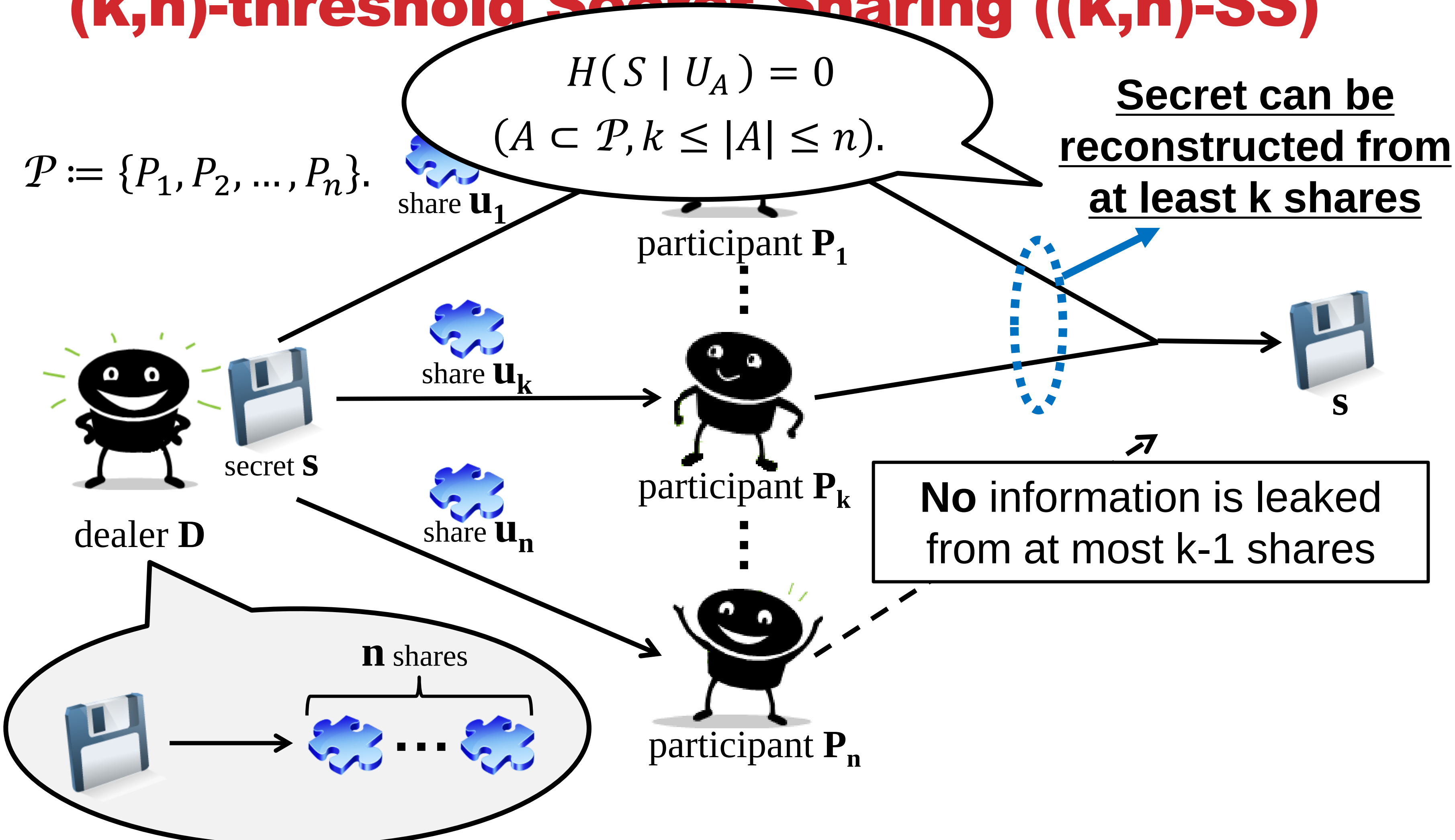


(k,n)-threshold Secret Sharing ((k,n)-SS)

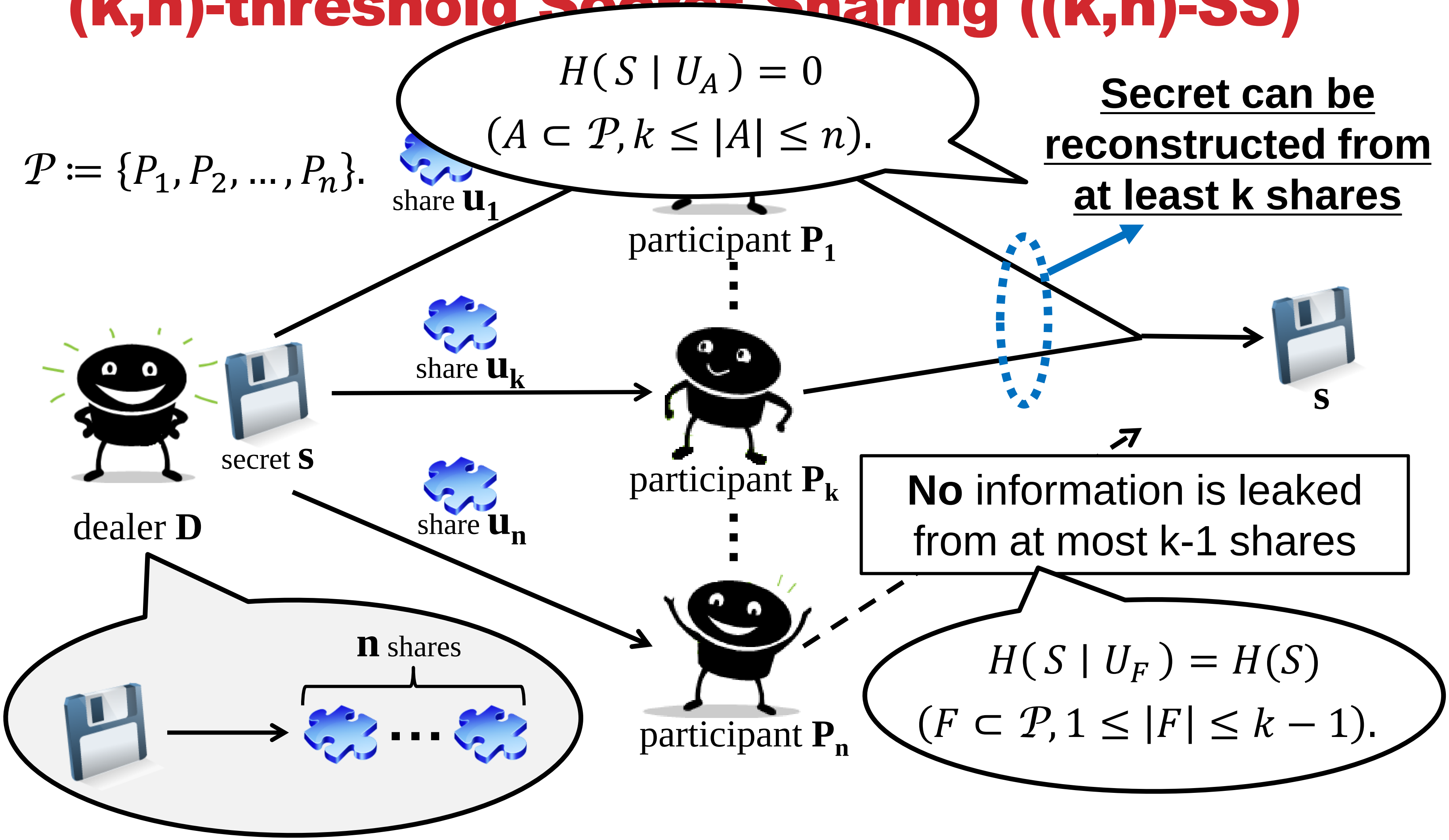
$\mathcal{P} := \{P_1, P_2, \dots, P_n\}.$



(k,n)-threshold Secret Sharing ((k,n)-SS)



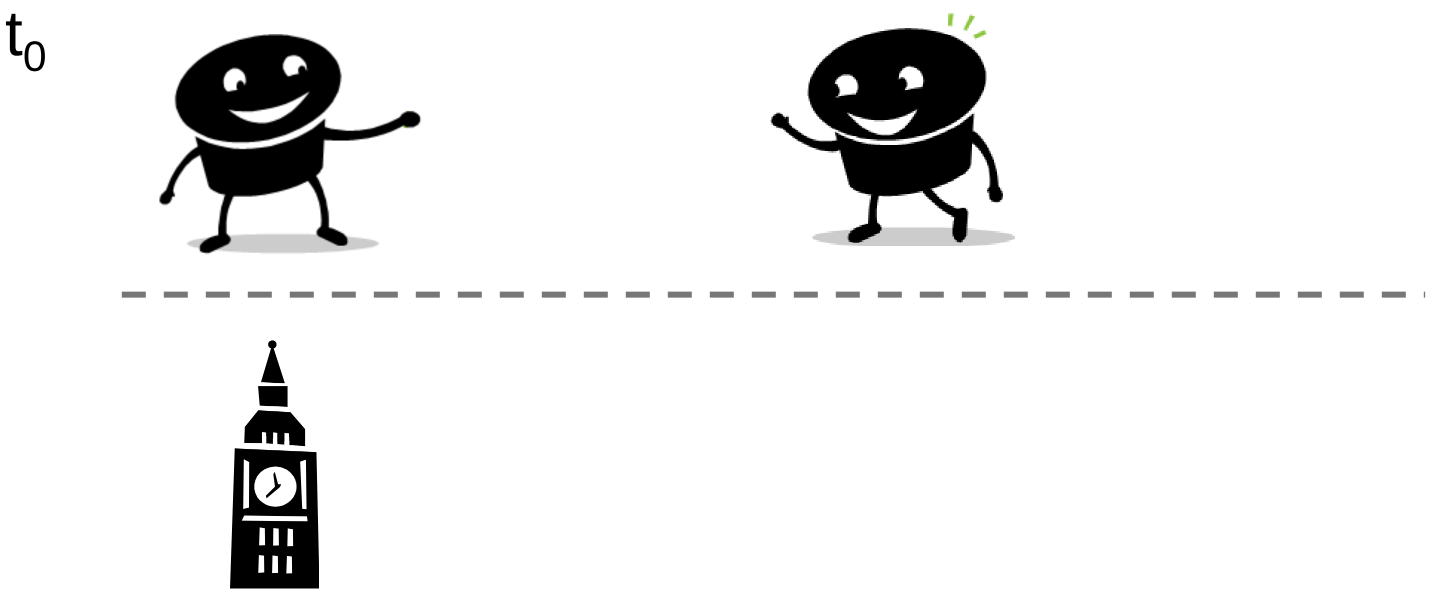
(k,n)-threshold Secret Sharing ((k,n)-SS)



Timed-Release Cryptography

Goal: securely send certain information into the future.

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

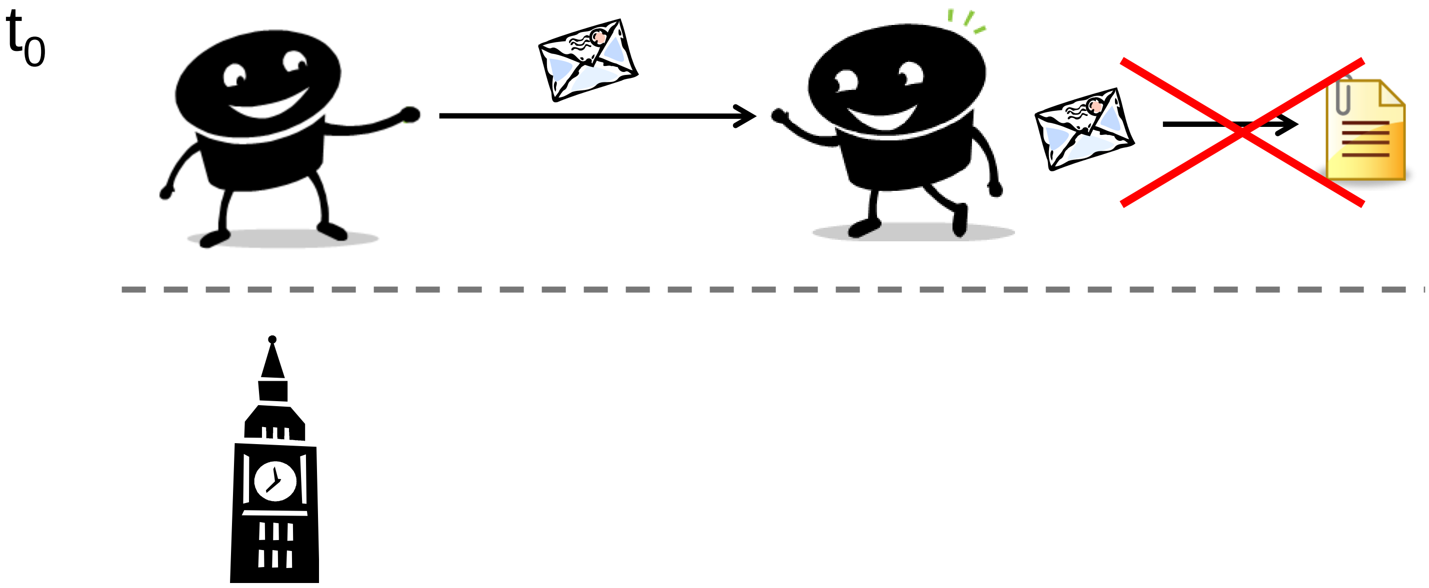


[RSW96] R. Rivest, A. Shamir, D.A. Wagner, “Time-lock puzzles and timed-release crypto”MIT LCS Tech. Report. MIT LCS TR-684,1996.

Timed-Release Cryptography

Goal: securely send certain information into the future.

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

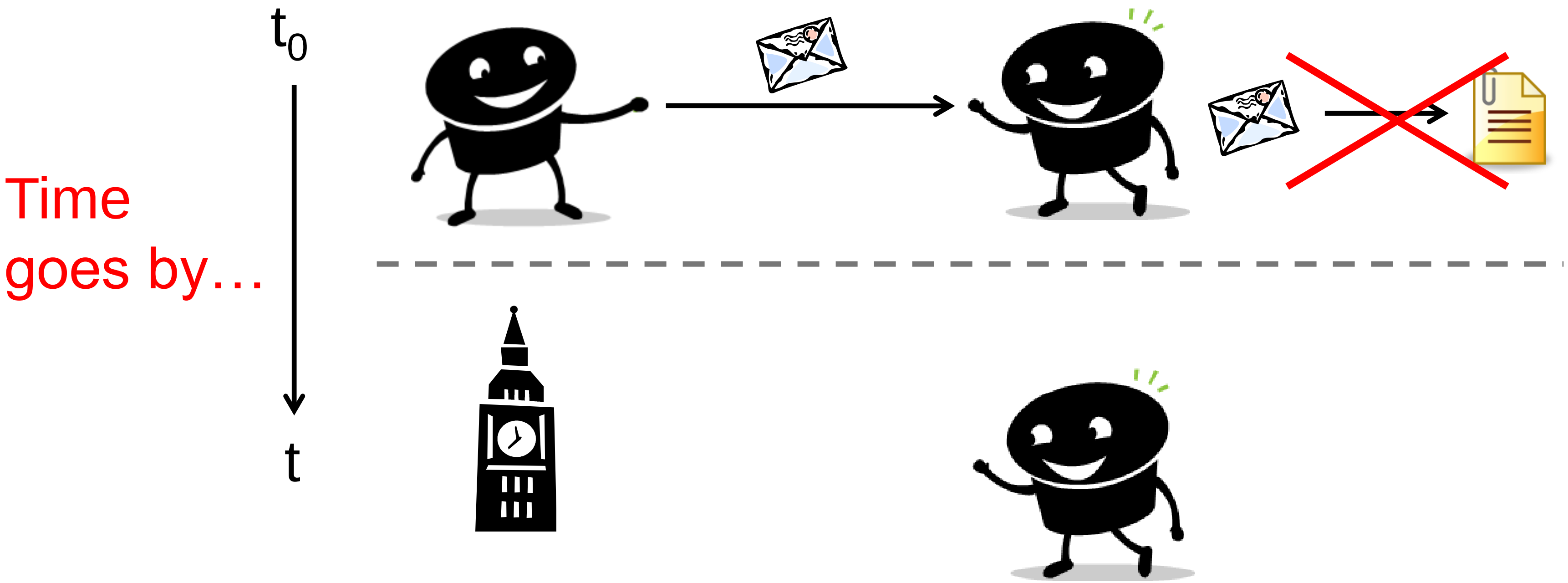


[RSW96] R. Rivest, A. Shamir, D.A. Wagner, “Time-lock puzzles and timed-release crypto”MIT LCS Tech. Report. MIT LCS TR-684,1996.

Timed-Release Cryptography

Goal: securely send certain information into the future.

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

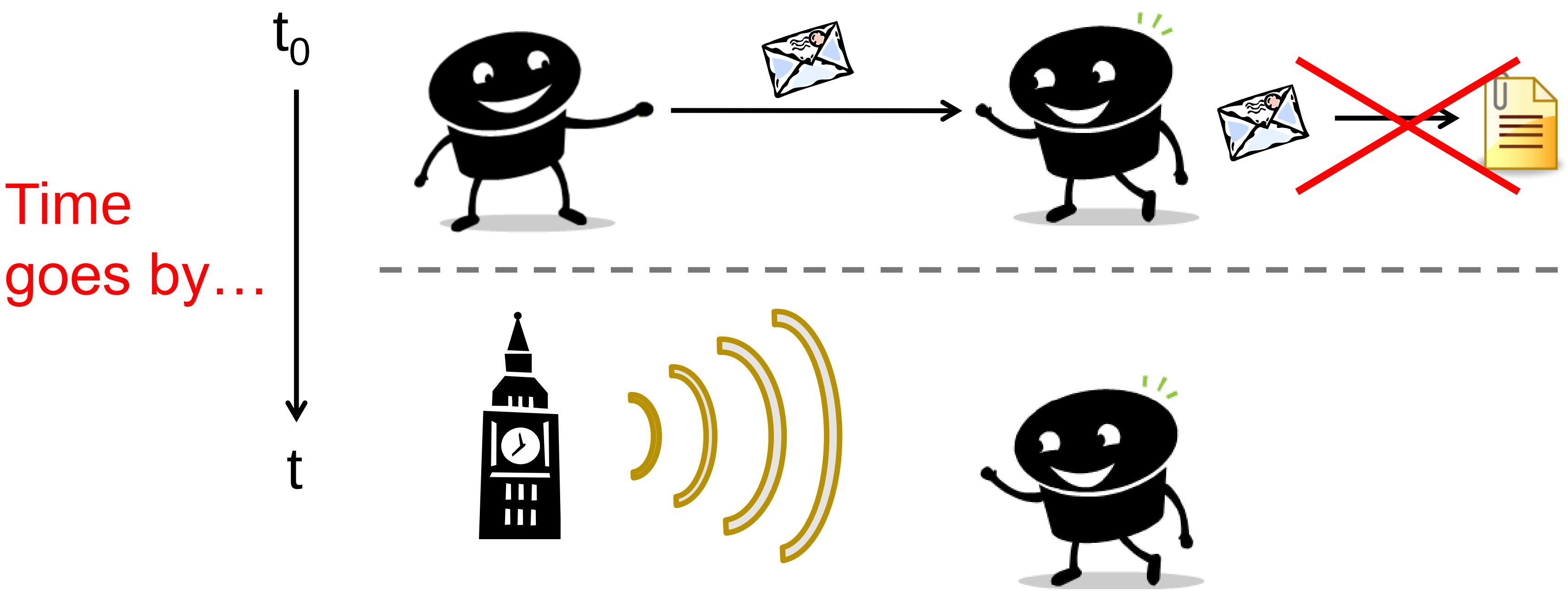


[RSW96] R. Rivest, A. Shamir, D.A. Wagner, "Time-lock puzzles and timed-release crypto"MIT LCS Tech. Report. MIT LCS TR-684,1996.

Timed-Release Cryptography

Goal: securely send certain information into the future.

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

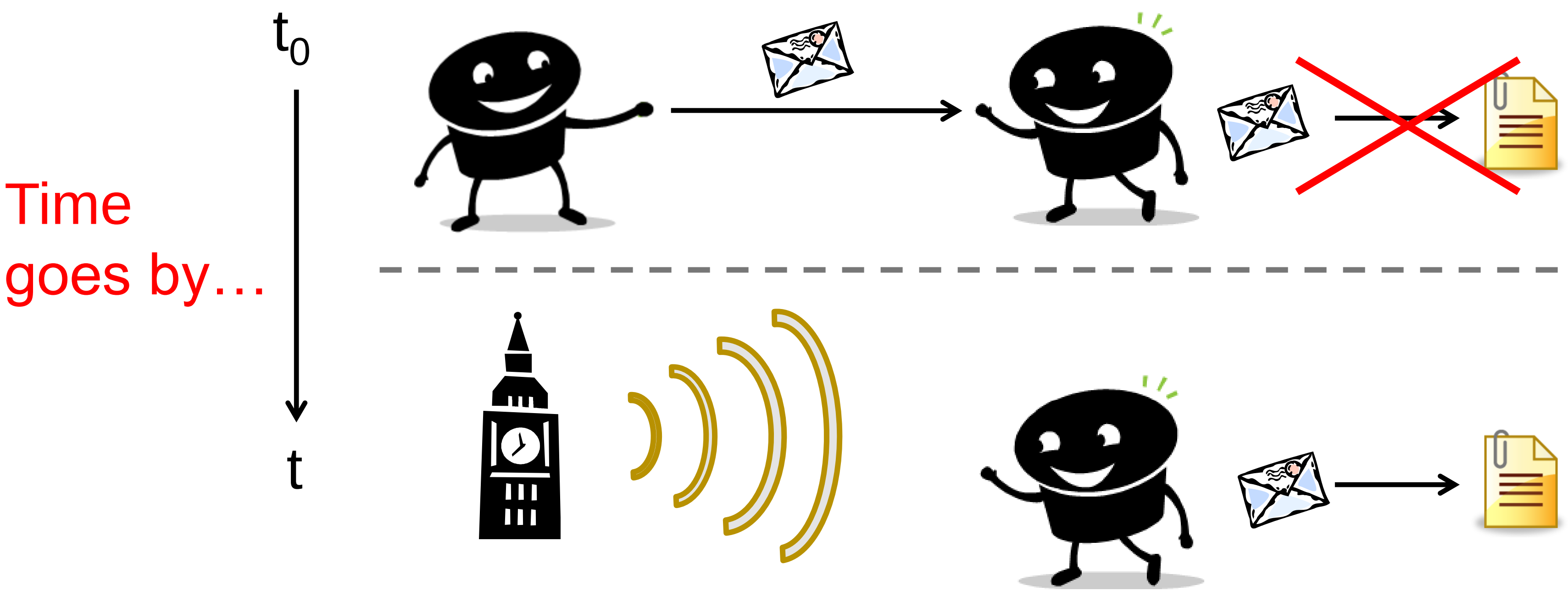


[RSW96] R. Rivest, A. Shamir, D.A. Wagner, "Time-lock puzzles and timed-release crypto"MIT LCS Tech. Report. MIT LCS TR-684,1996.

Timed-Release Cryptography

Goal: securely send certain information into the future.

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



[RSW96] R. Rivest, A. Shamir, D.A. Wagner, “Time-lock puzzles and timed-release crypto”MIT LCS Tech. Report. MIT LCS TR-684,1996.

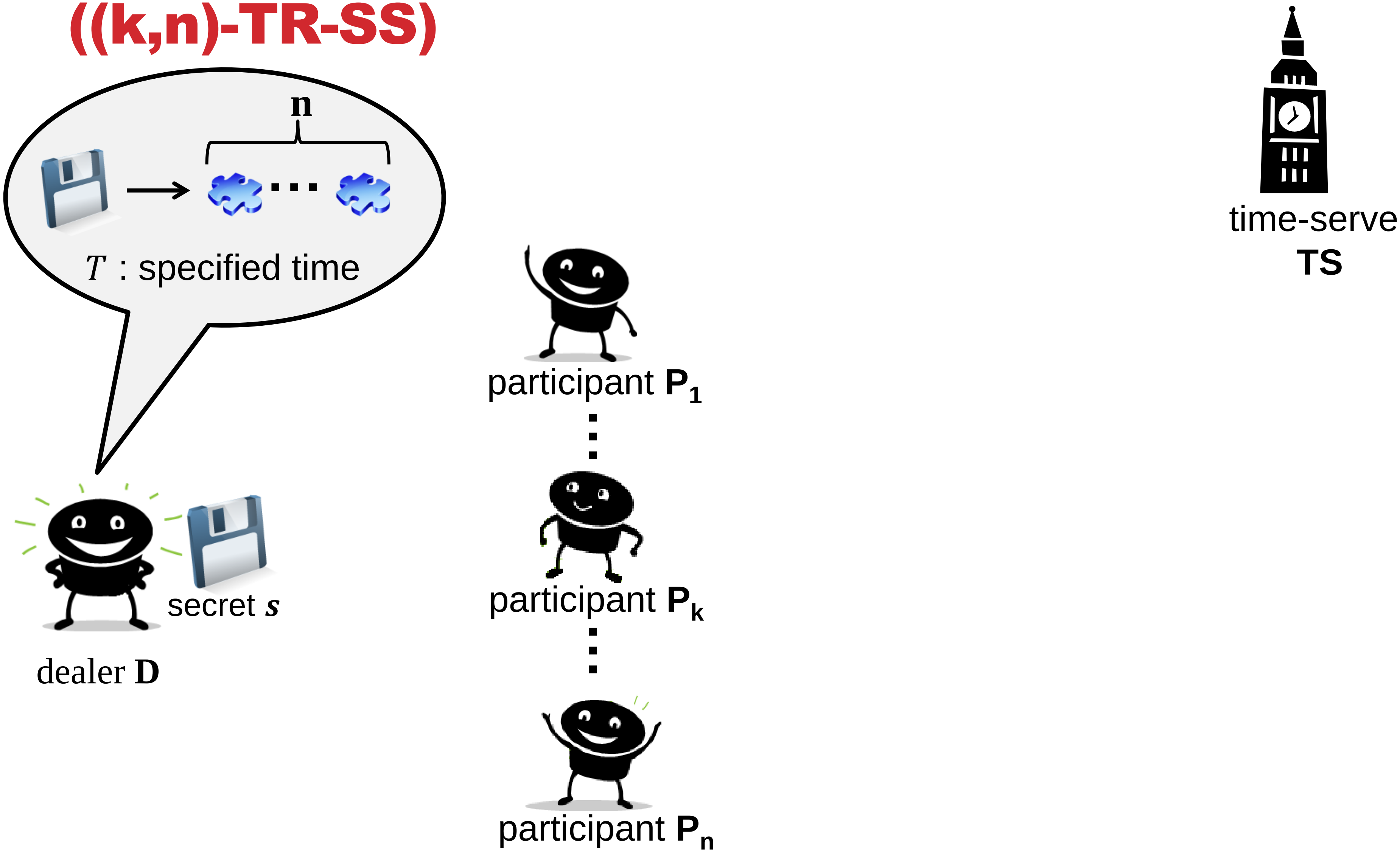
Our Proposal

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

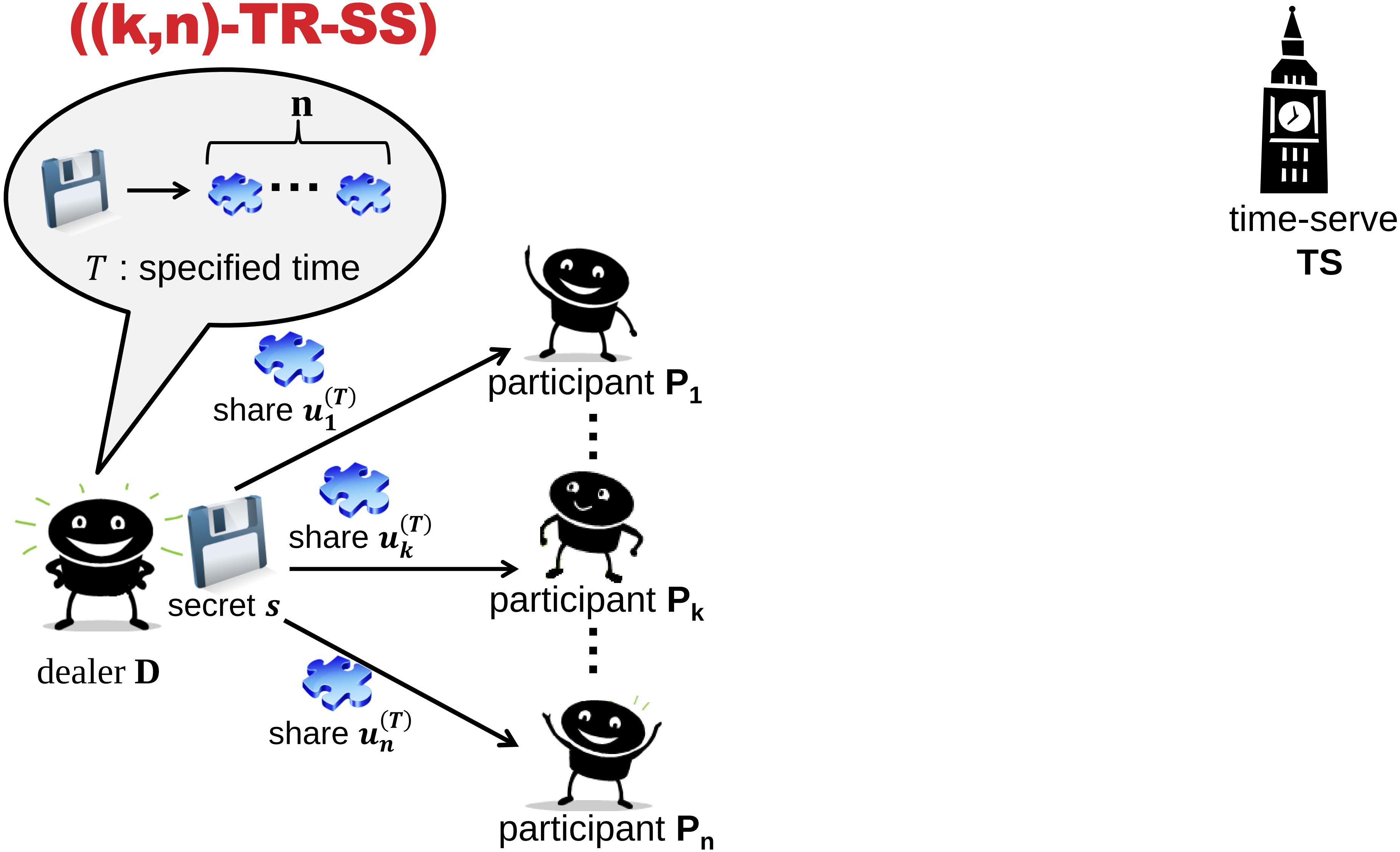
- (k, n)**-TR-SS: Realize reconstruction with timed-release functionality.**
  - Formalize a model and security notions.
  - Derive lower bounds on sizes of shares, time-signals and secret keys.
  - Propose an optimal direct construction in the sense that it meets equality in the above every bound.

- (k_1, k_2, n)**-TR-SS: Realize timed-release functionality and secret sharing functionality *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.

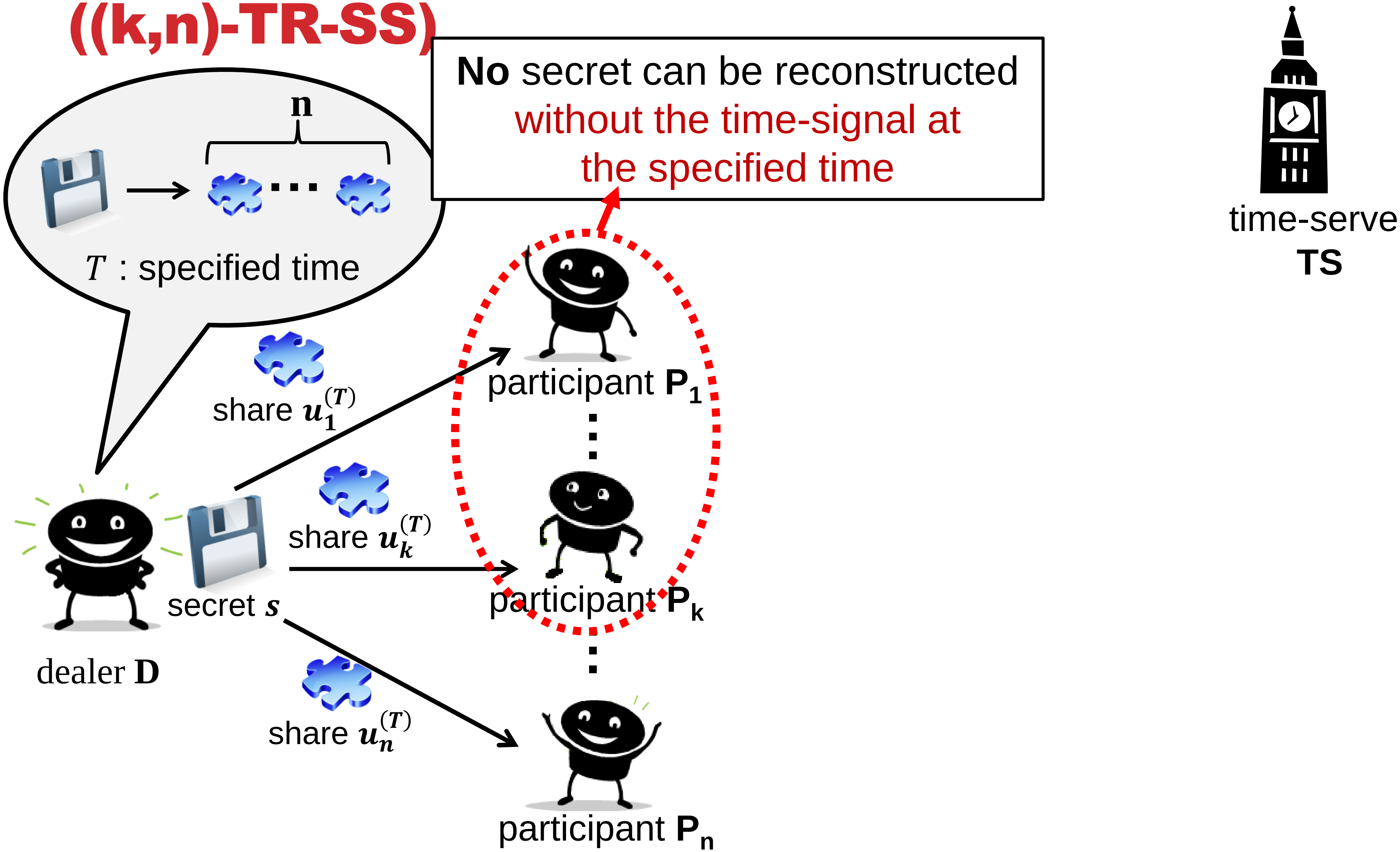
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)



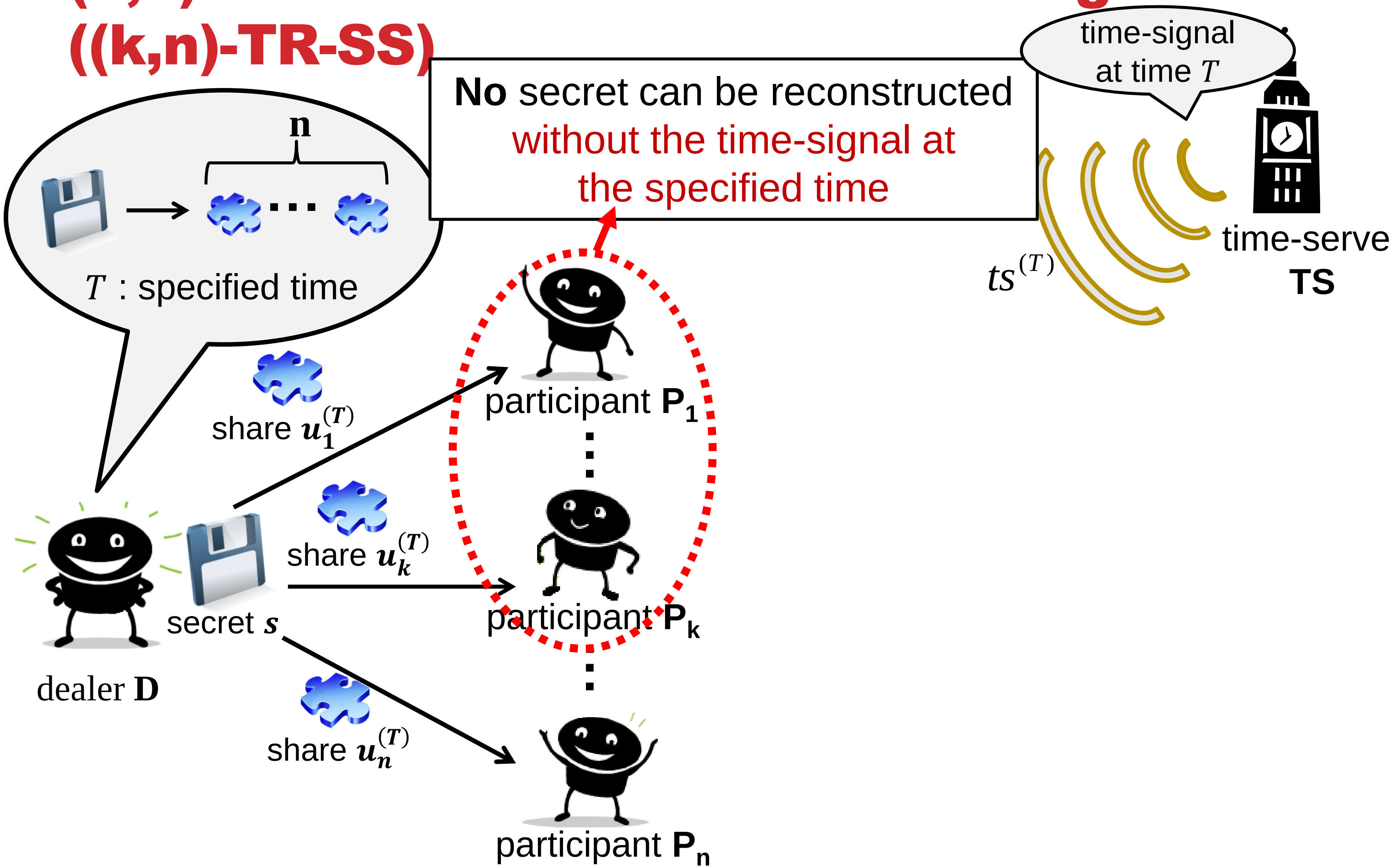
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)



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

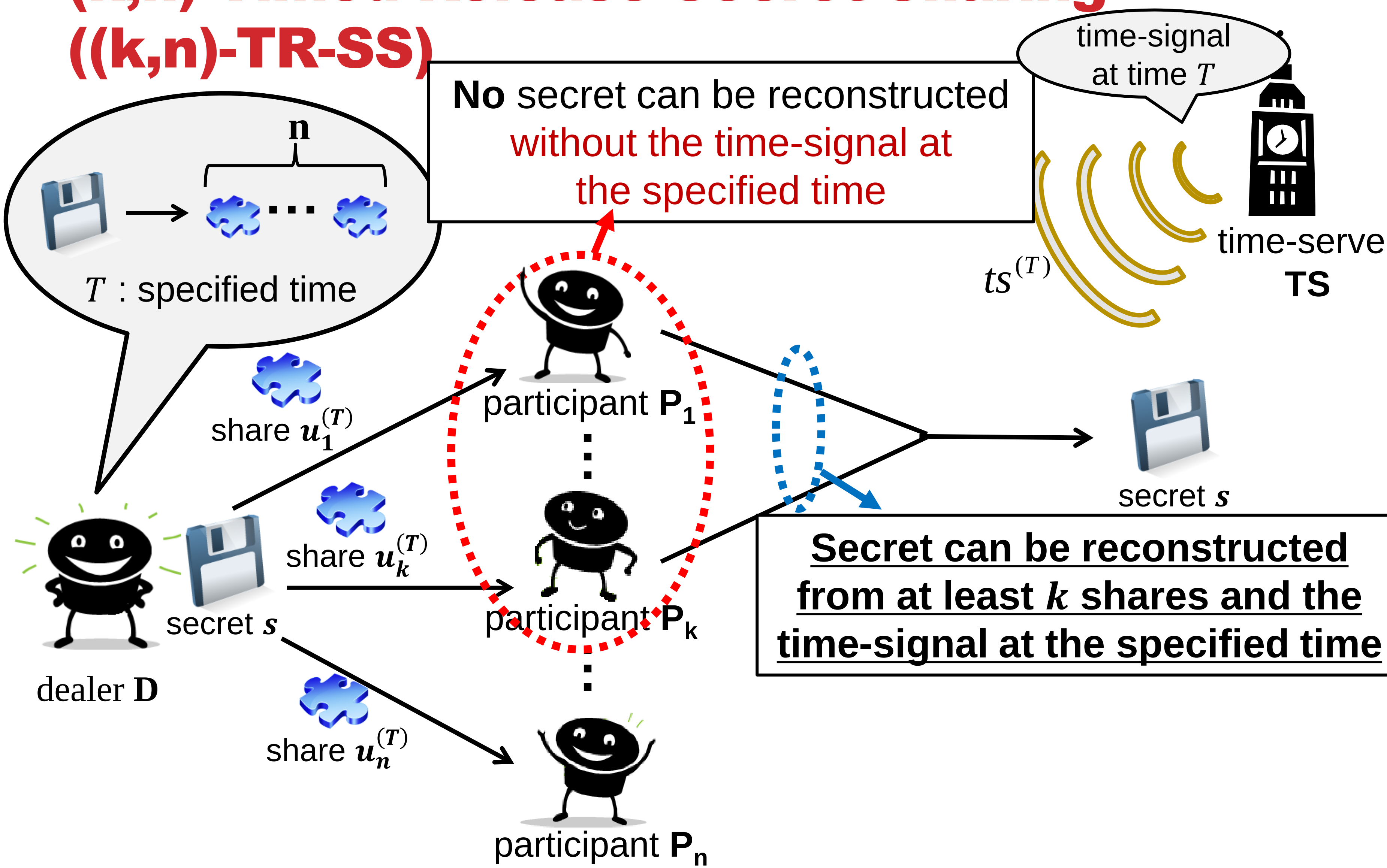


(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)



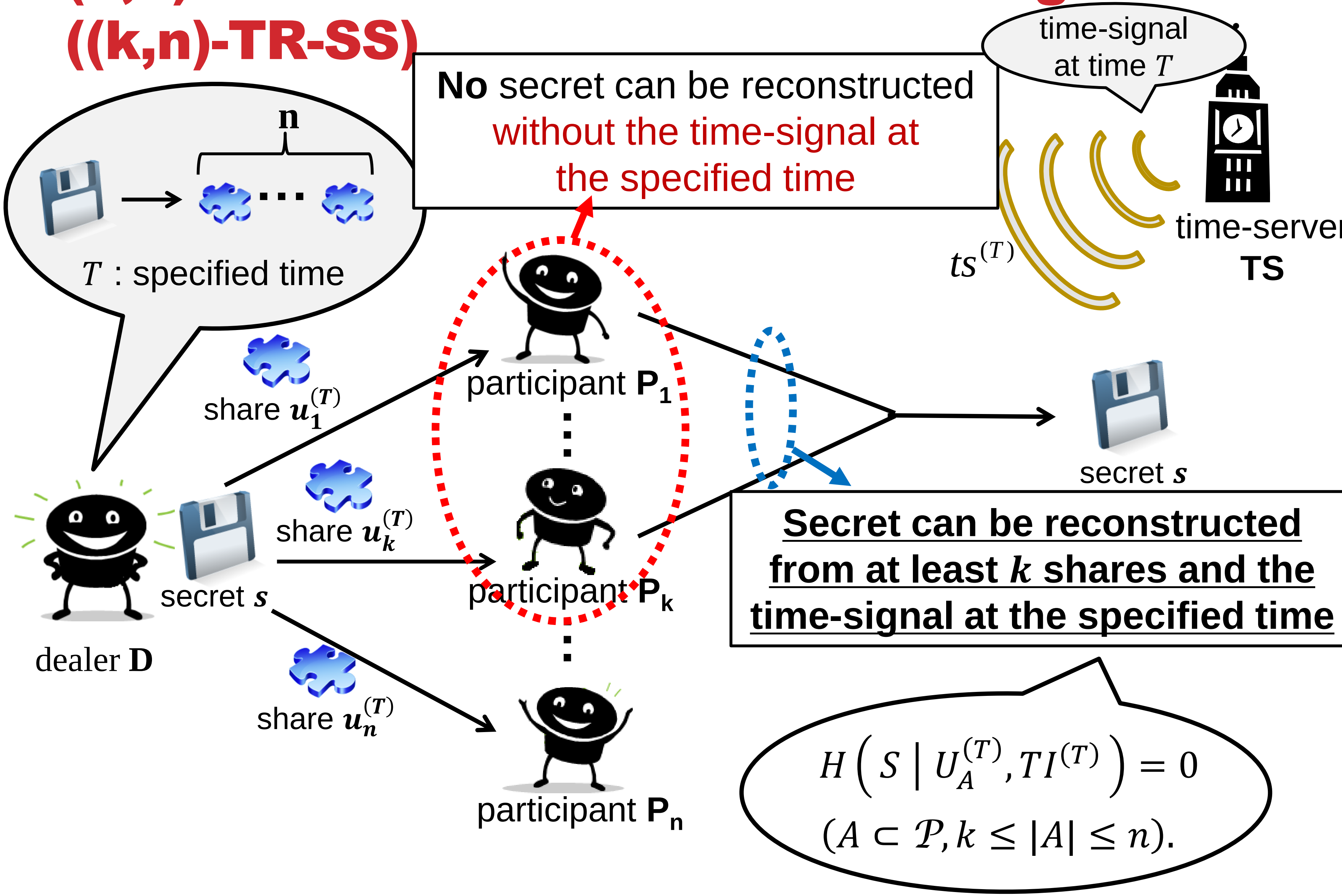
(k,n)-Timed-Release Secret Sharing ((k,n)-TR-SS)

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

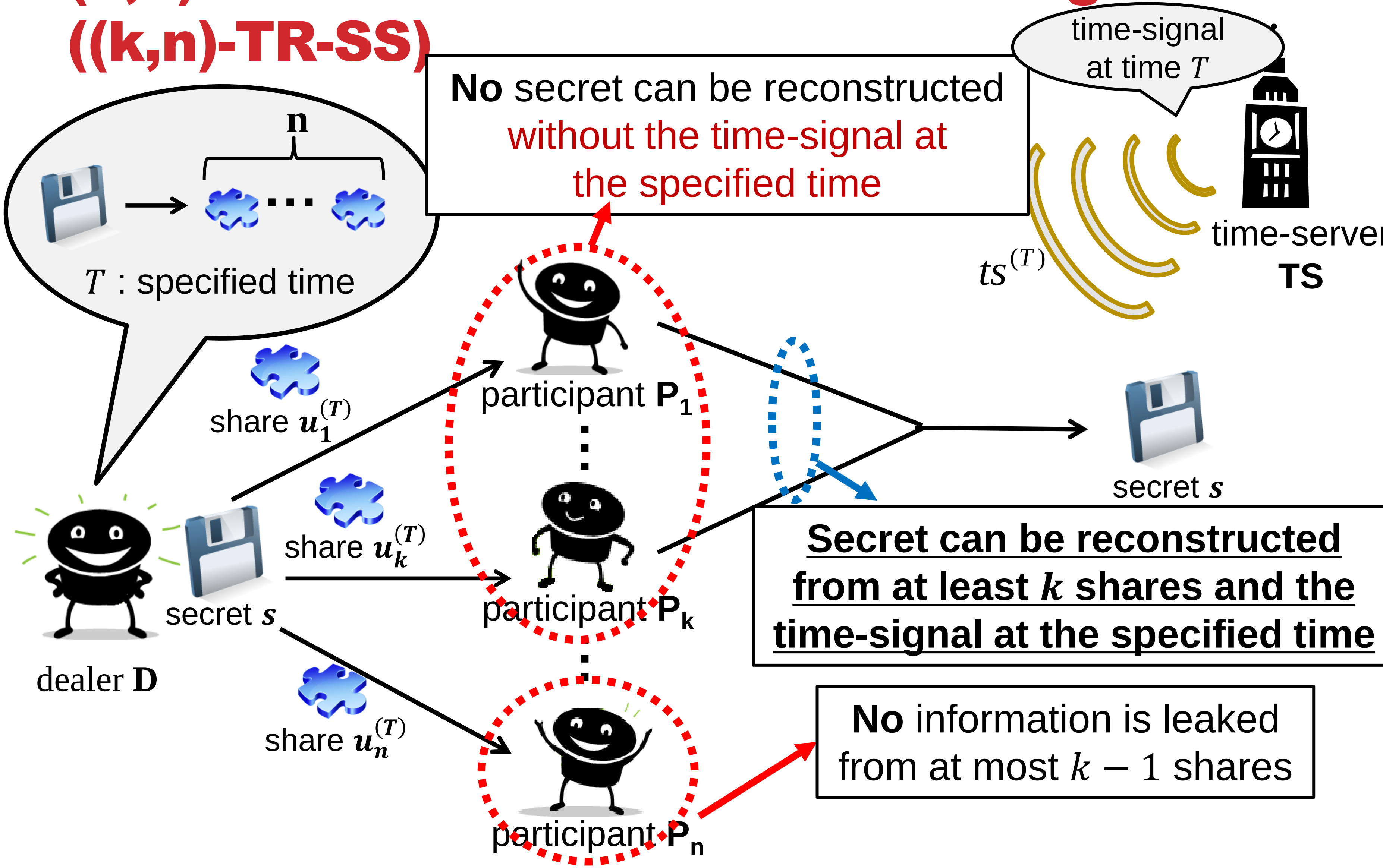


Secret can be reconstructed from at least k shares and the time-signal at the specified time

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



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



(k,n)-TR-SS: Model

Entities.

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

Phases.

Initialize, Share, Extract and *Reconstruct*.

Spaces.

$\mathcal{S}$: a set of secrets;

$\mathcal{SK}$: a set of secret keys;

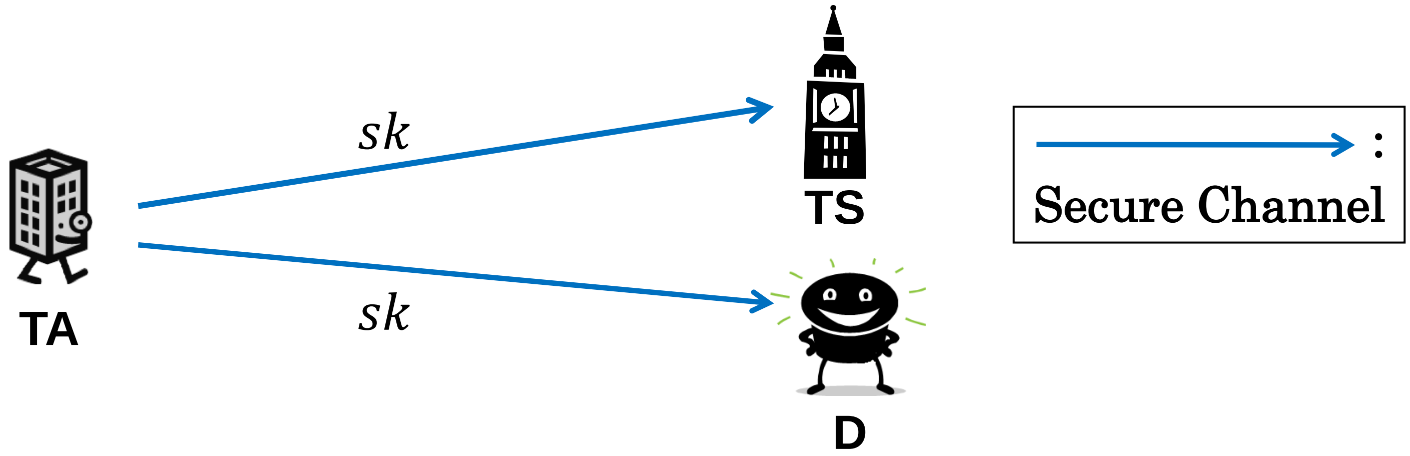
$\mathcal{T} := \{1,2,\ldots,\tau\}$: a set of time;

$\mathcal{U}$: a set of shares, where $\mathcal{U} := \bigcup_{i=1}^{n} \mathcal{U}_i$ and $\mathcal{U}_i := \bigcup_{t=1}^{\tau} \mathcal{U}_i^{(t)}$;

$\mathcal{TI}$: a set of time-signals, where $\mathcal{TI} := \bigcup_{t=1}^{\tau} \mathcal{TI}^{(t)}$.

(k,n)-TR-SS: Model

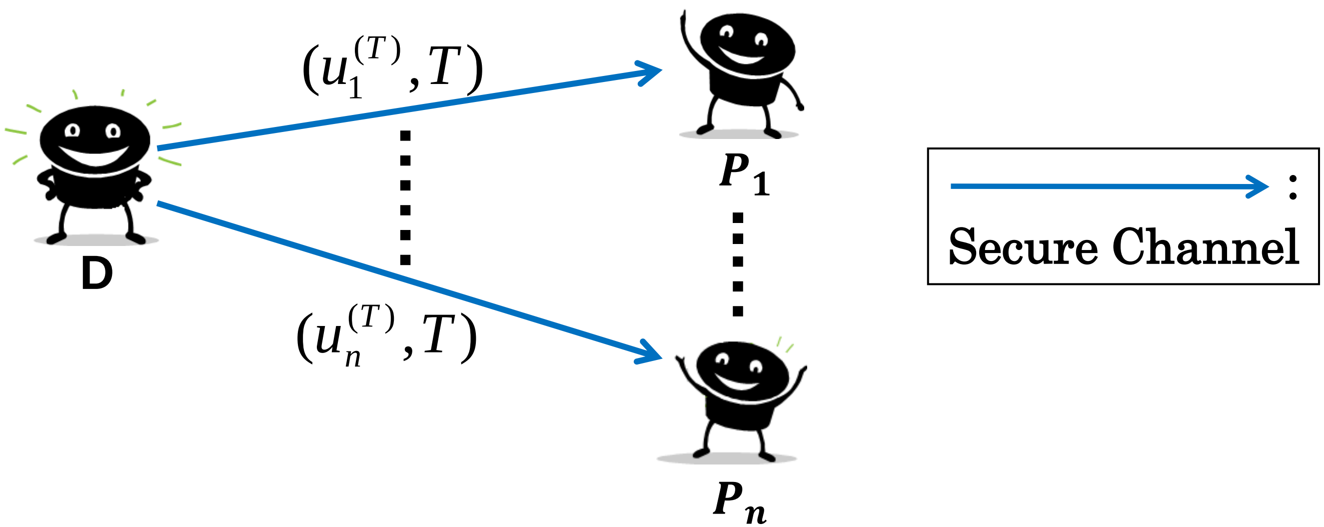
1. **Initialize.**
   1. **TA** generates a secret key $sk \in \mathcal{SK}$ for **TS** and **D**.
   2. **TA** distributes sk to **TS** and **D** via secure channels.
   3. **TA** deletes sk from his memory.



(k,n)-TR-SS: Model

2. <u>Share.</u>

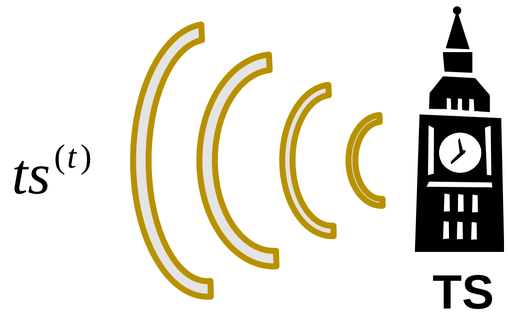
   1. **D** randomly selects a secret $s \in \mathcal{S}$ and chooses k and n.
   2. **D** specifies future time $T \in \mathcal{T}$, and computes n shares $u_1^{(T)}, \ldots, u_n^{(T)}$.
   3. **D** sends $\left(u_i^{(T)}, T\right)$ to $\boldsymbol{P_i}$ via a secure channel $(i = 1,2, \ldots, n)$.



(k,n)-TR-SS: Model

3. **Extract.**

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

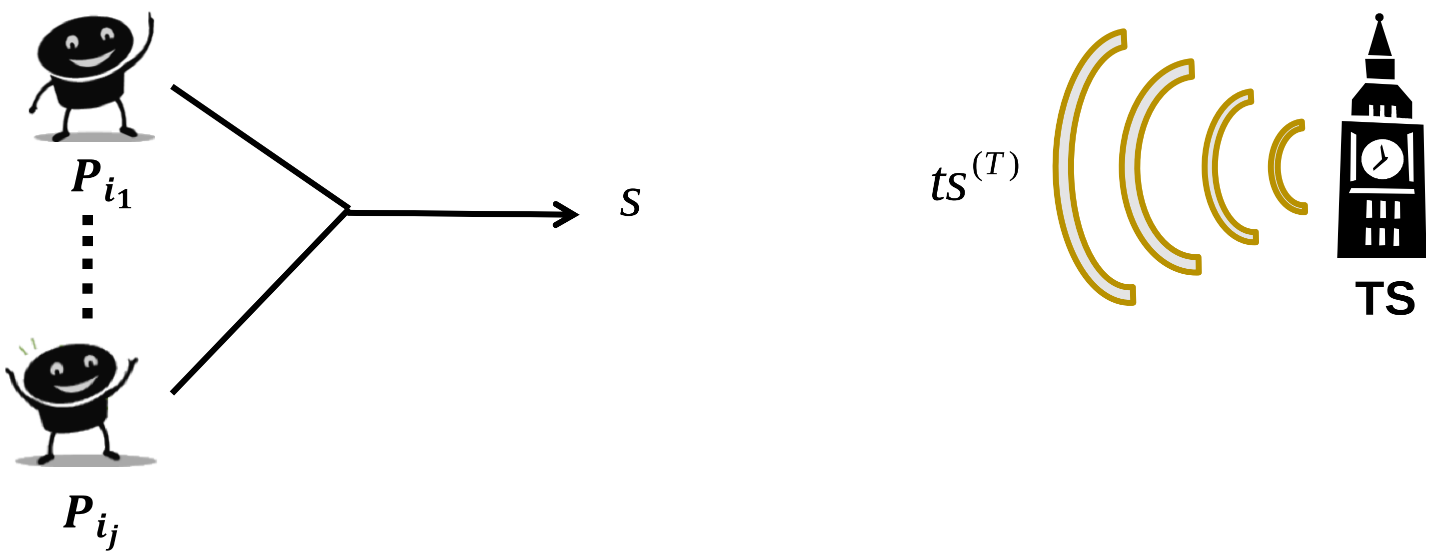


For simplicity, we assume $ts^{(t)}$ is deterministically computed by t and sk.

(k,n)-TR-SS: Model

4. **Reconstruct.**

At the specified time T, any set of participants $A := \{P_{i_1}, \dots, P_{i_j}\}$ $(k \leq j \leq n)$ can reconstruct s from their shares $u_{i_1}^{(T)}, \dots, u_{i_j}^{(T)}$ and a time-signal $ts^{(T)}$ 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 $F \subset \mathcal{P}$ s.t. $1 \leq |F| \leq k-1$ and any $T \in \mathcal{T}$, it holds that

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

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

$$H\left(S \mid U_A^{(T)}, TI^{(1)}, \ldots, TI^{(T-1)}, TI^{(T+1)}, \ldots, TI^{(\tau)}\right) = H(S).$$

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

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

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

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

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

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.

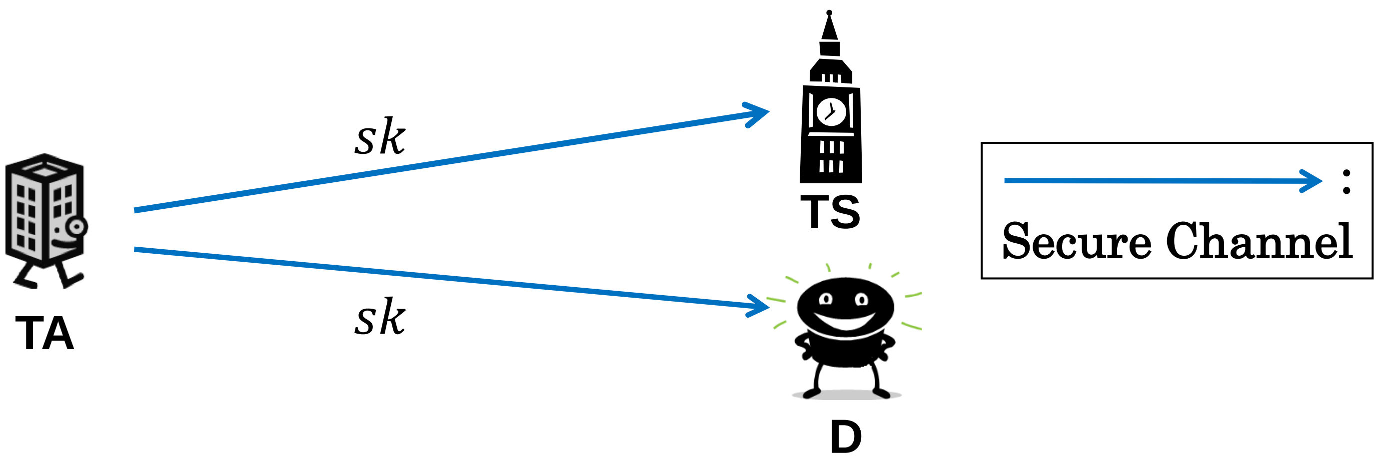
(k,n)-TR-SS: Optimal Construction

1. **Initialize.**

   Let q be a prime power, where $q > \max(n, \tau)$.

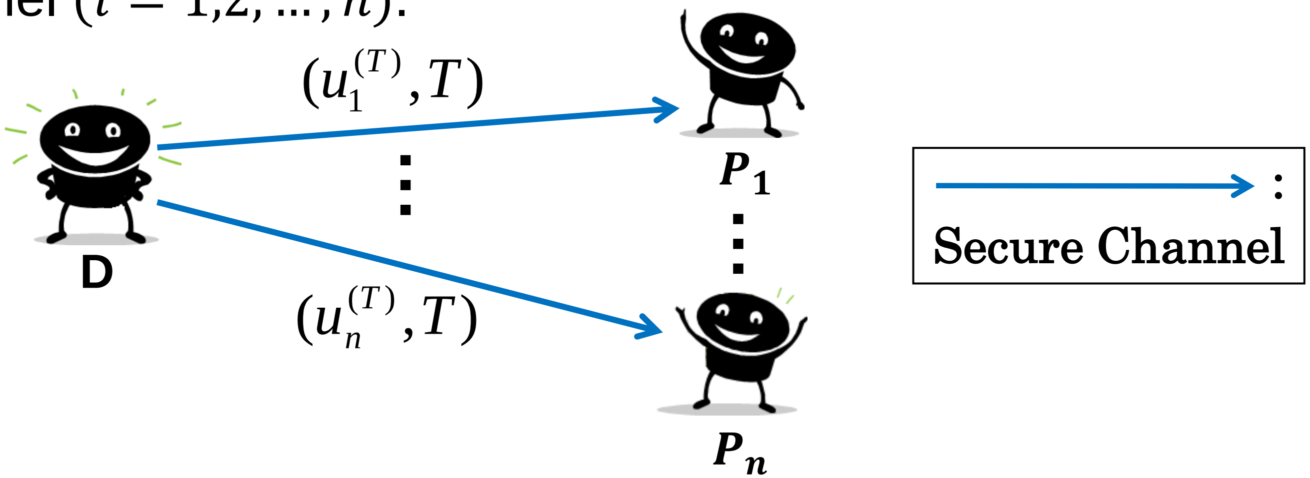
   Let $\mathbf{F}_q$ be a finite field with q elements.

   1. **TA** chooses τ numbers $r^{(j)}$ $(j = 1, \dots, \tau)$ from $\mathbf{F}_q$ uniformly at random.
   2. **TA** sends $sk := (r^{(1)}, \dots, r^{(\tau)})$ to **TS** and **D**, respectively.



(k,n)-TR-SS: Optimal Construction

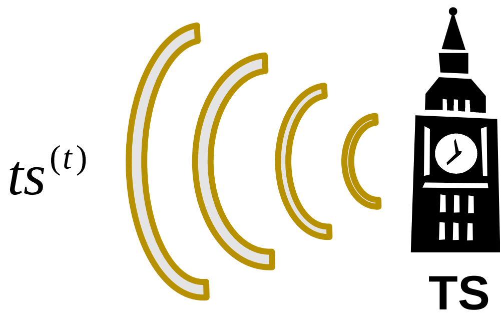
2. **Share.**
   1. **D** randomly selects a secret $s \in \mathbf{F}_q$ and chooses k and n.
   2. **D** specifies future time $T \in \mathcal{T}$.
   3. **D** randomly chooses $f(x) \coloneqq c^{(T)} + \sum_{i=1}^{k-1} a_i x^i$ over $\mathbf{F}_q$, where $c^{(T)} \coloneqq s + r^{(T)}$ and each a_i is chosen from $\mathbf{F}_q$ uniformly at random.
   4. **D** computes $u_i^{(T)} \coloneqq f(P_i)$ and sends $\left(u_i^{(T)}, T\right)$ to $\boldsymbol{P_i}$ via a secure channel $(i = 1,2, \dots, n)$.



(k,n)-TR-SS: Optimal Construction

3. **Extract.**

At each time $t \in \mathcal{T}$, **TS** broadcasts t-th key $r^{(t)}$ as a time-signal at time t.



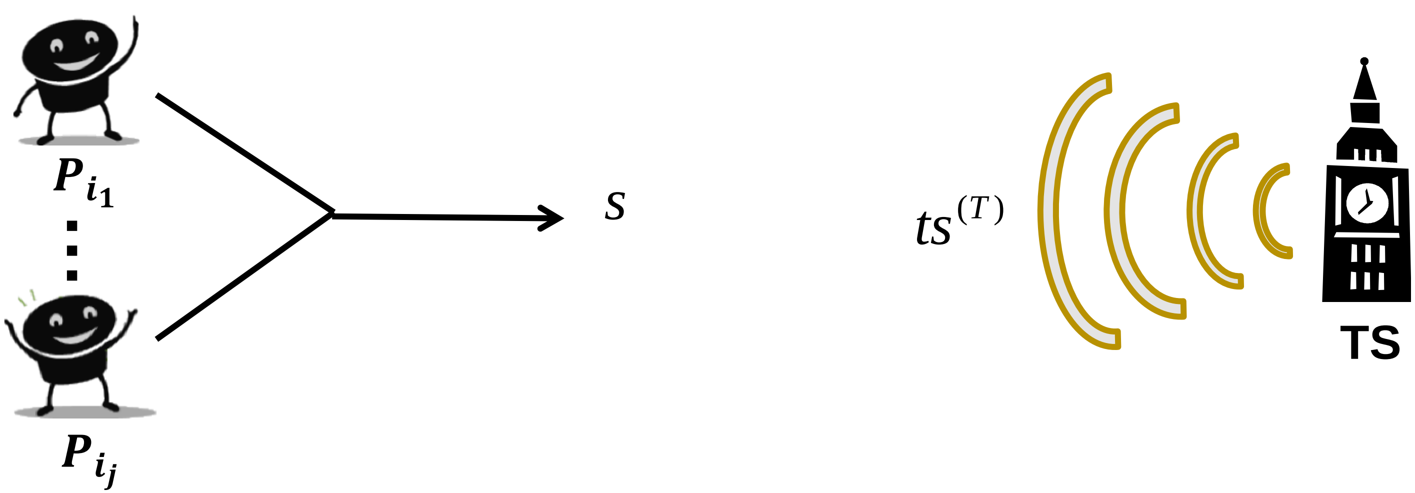
(k,n)-TR-SS: Optimal Construction

4. **Reconstruct.**

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

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

   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 $A \coloneqq \left\{P_{i_1}, \ldots, P_{i_j}\right\}$ can compute $c^{(T)}$ by Lagrange interpolation from their k shares:
   $$c^{(T)} = \sum_{j=1}^{k} \left(\prod_{l \neq j} \frac{P_{i_j}}{P_{i_j} - P_{i_l}} \right) f\left(P_{i_j}\right).$$
   2. After receiving $ts^{(T)} = r^{(T)}$, they can compute $s = c^{(T)} - r^{(T)}$.

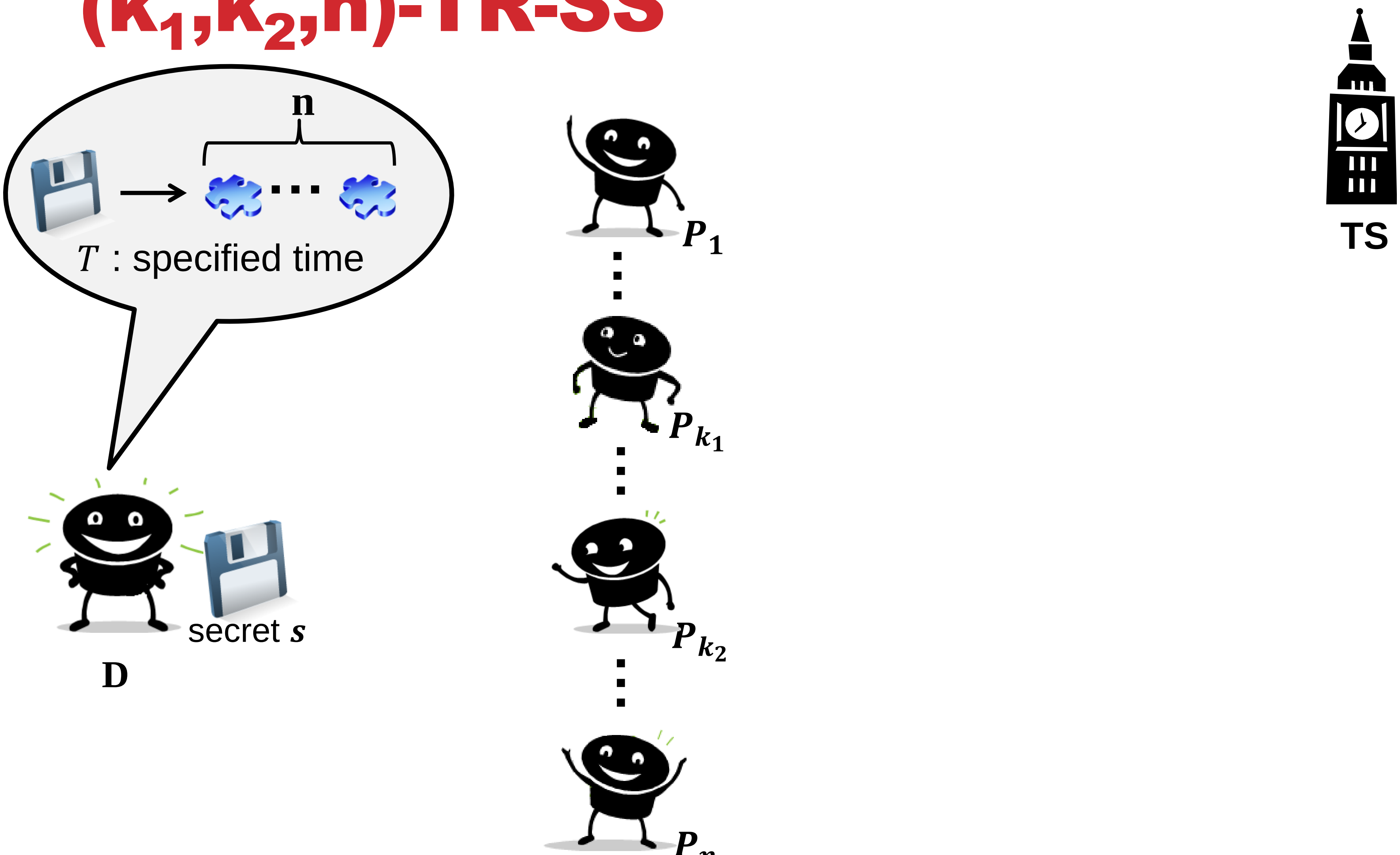
P_{i_j}

<u>Theorem.</u>

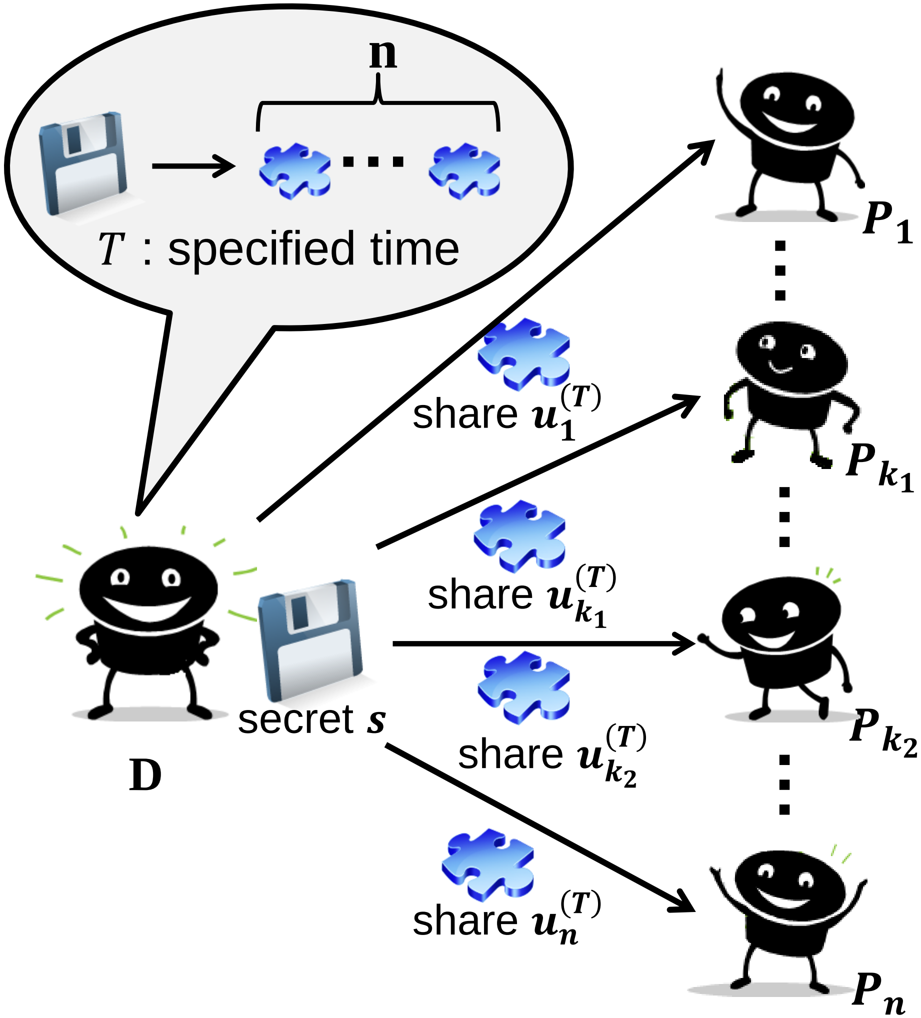
The resulting (k,n)-TR-SS scheme by this construction is *secure* and *optimal*.

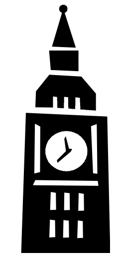
TS

(k_1,k_2,n)-TR-SS



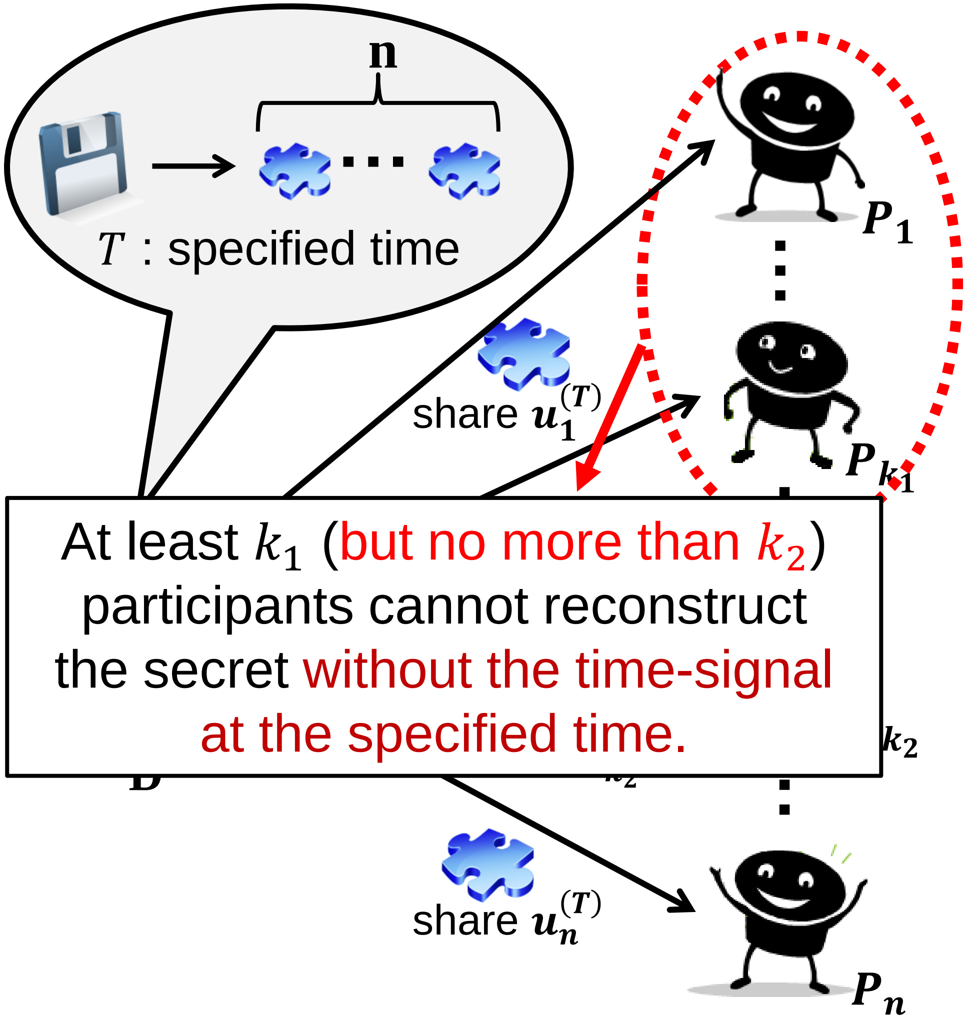
(k_1,k_2,n)-TR-SS

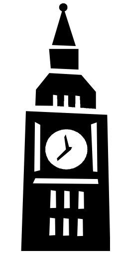




TS

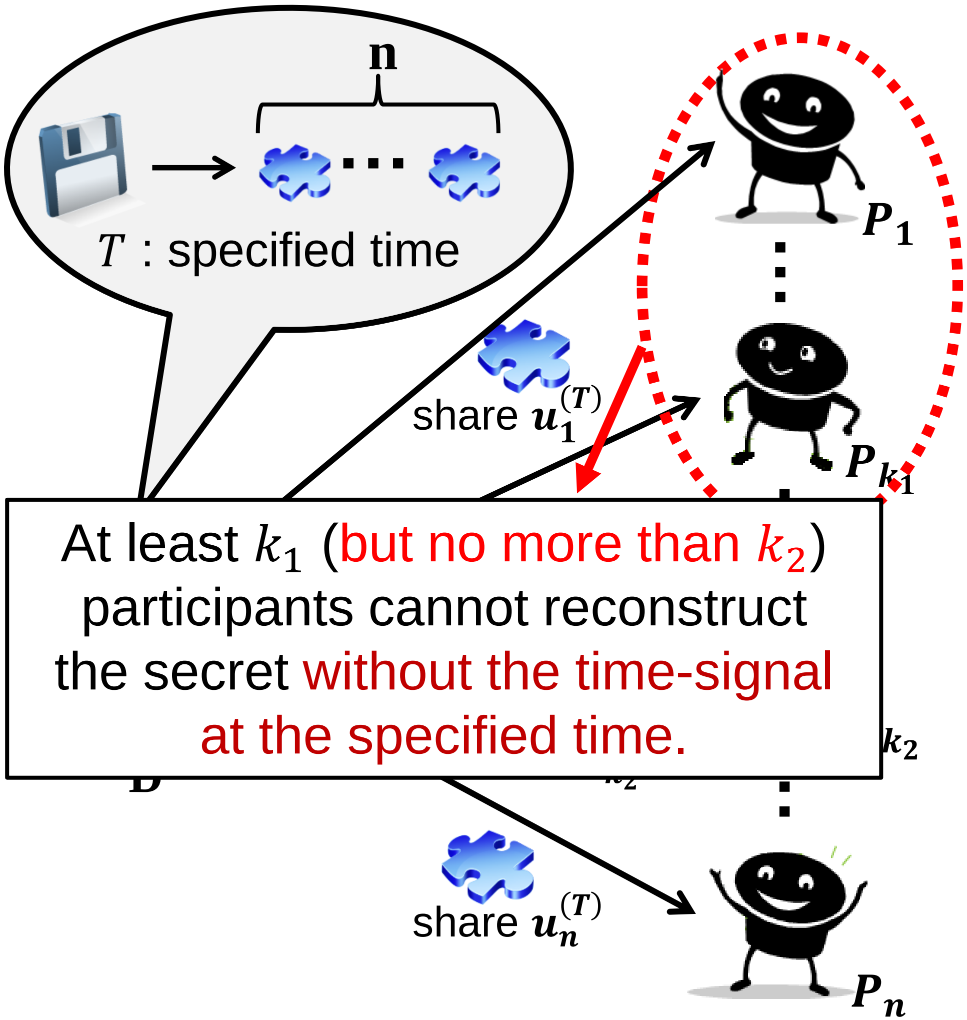
(k₁,k₂,n)-TR-SS



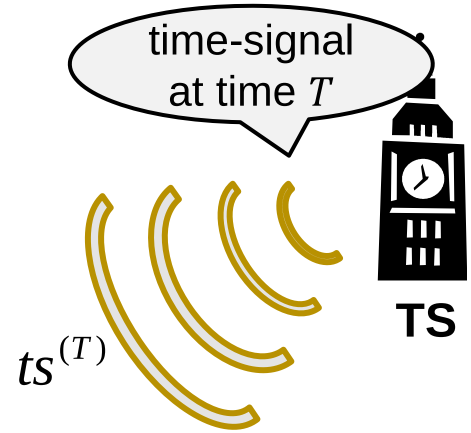


At least k_1 (but no more than k_2) participants cannot reconstruct the secret without the time-signal at the specified time.

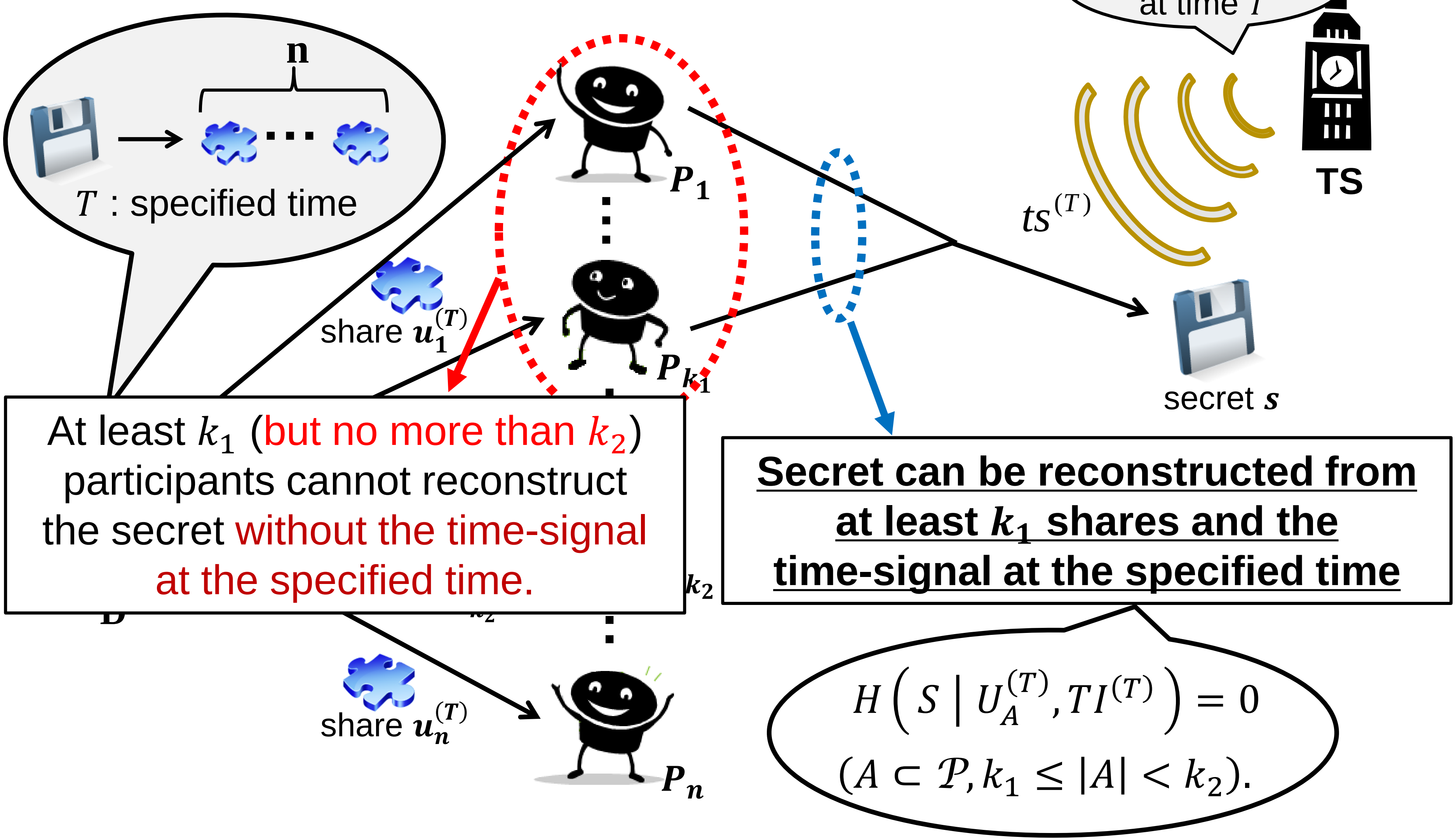
(k_1,k_2,n)-TR-SS



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



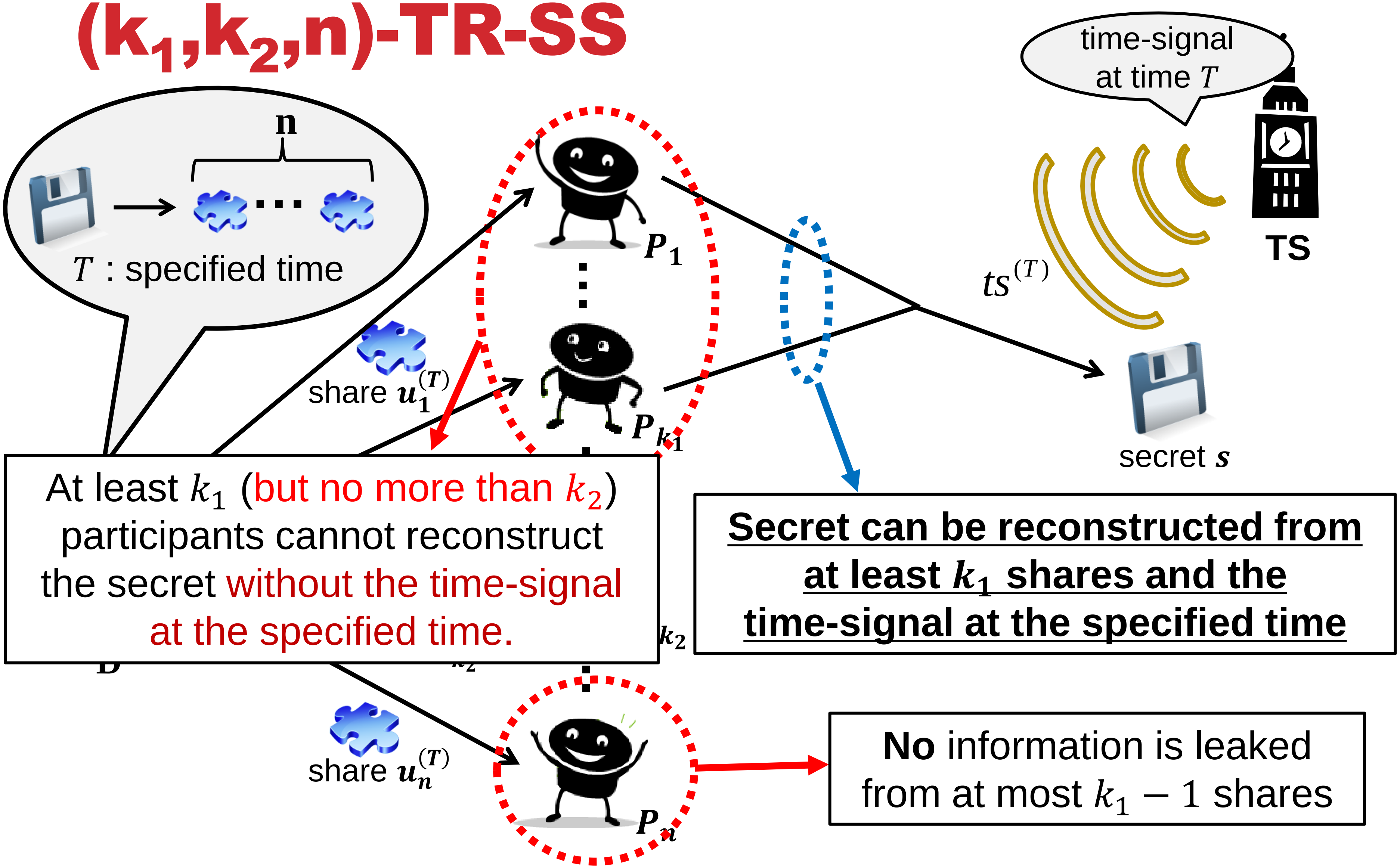
At least k_1 (but no more than k_2) participants cannot reconstruct the secret without the time-signal at the specified time.

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

$$H\left(S \mid U_A^{(T)}, TI^{(T)}\right) = 0$$

$$(A \subset \mathcal{P}, k_1 \leq |A| < k_2).$$

(k_1,k_2,n)-TR-SS

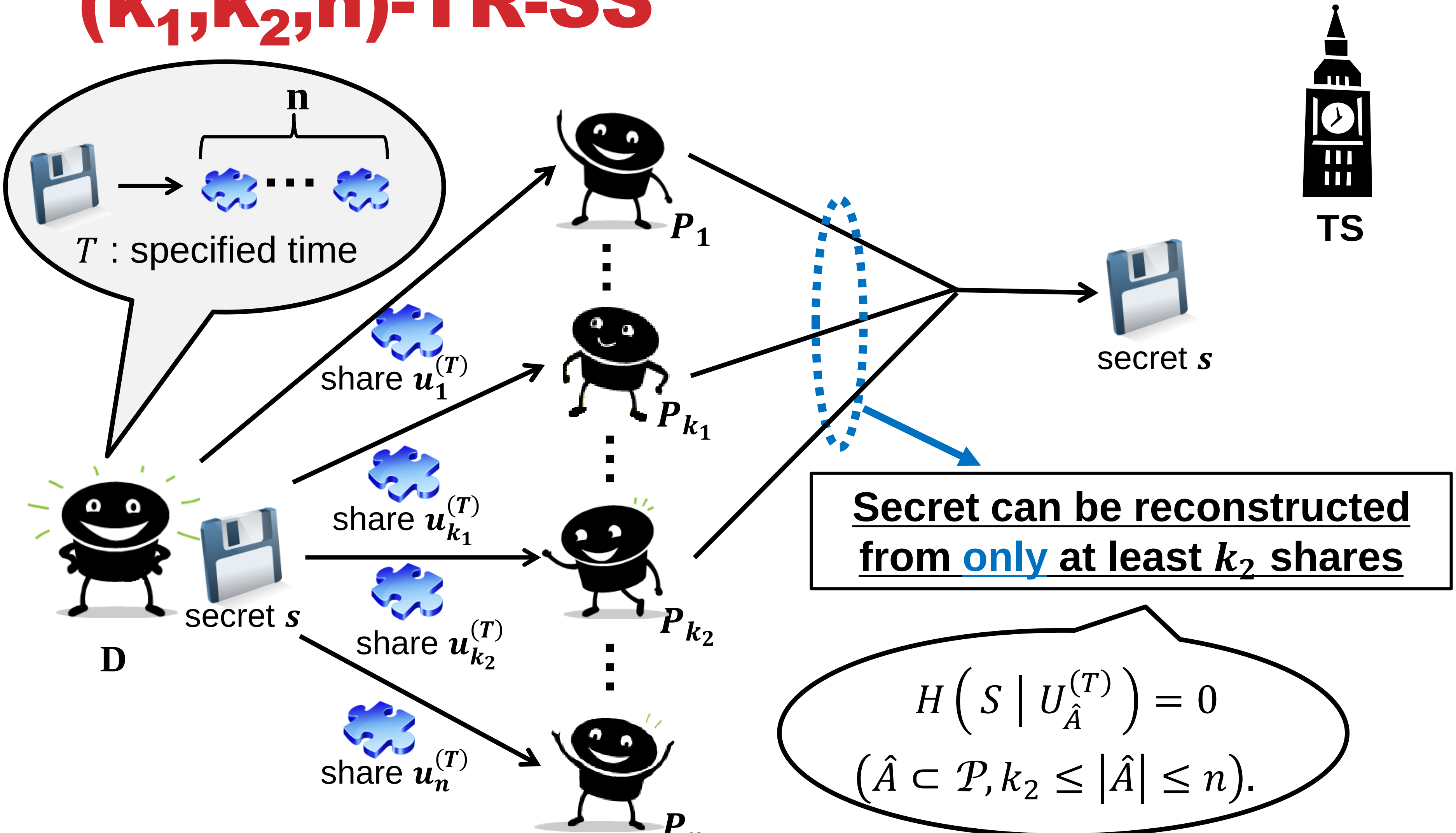


At least k_1 (but no more than k_2) participants cannot reconstruct the secret without the time-signal at the specified time.

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

(k_1,k_2,n)-TR-SS



n
T : specified time
TS
P_1
P_{k_1}
P_{k_2}
P_n
D
secret s
share $u_1^{(T)}$
share $u_{k_1}^{(T)}$
share $u_{k_2}^{(T)}$
share $u_n^{(T)}$
secret s
Secret can be reconstructed from only at least k_2 shares
$H\left(S \mid U_{\hat{A}}^{(T)}\right) = 0$
$\left(\hat{A} \subset \mathcal{P}, k_2 \leq |\hat{A}| \leq n\right)$.

(k_1,k_2,n)-TR-SS: Model

Entities.

A dealer **D**, n participants $\mathcal{P} \coloneqq \{\boldsymbol{P_1}, \ldots \boldsymbol{P_n}\}$, a time-server **TS**, and a trusted authority **TA**.

Phases.

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

Spaces.

$\mathcal{S}$: a set of secrets;

$\mathcal{SK}$: a set of secret keys;

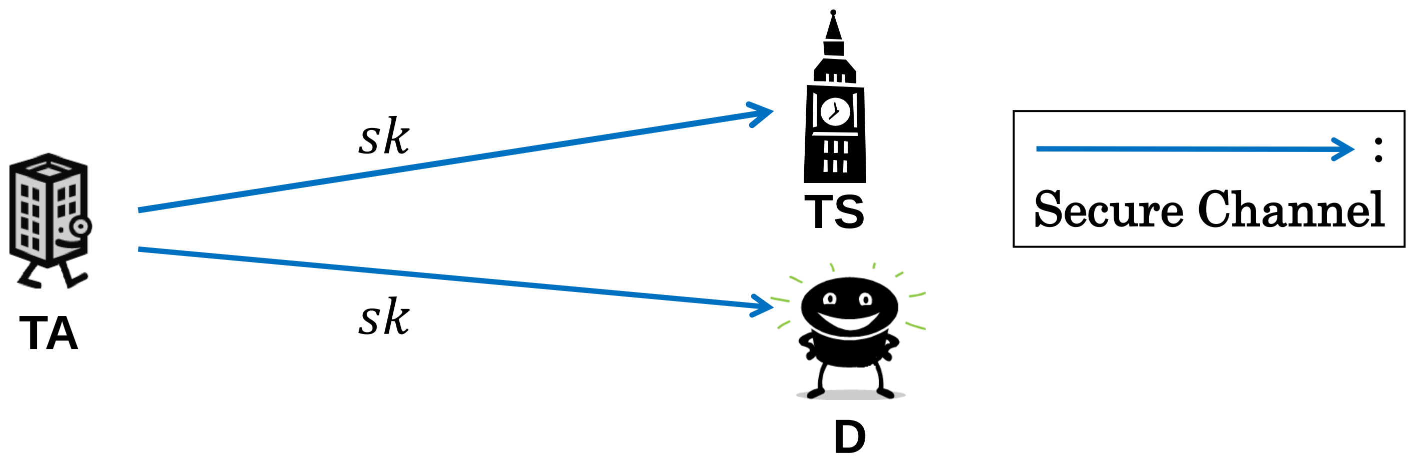
$\mathcal{T} \coloneqq \{1,2, \ldots, \tau\}$: a set of time;

$\mathcal{U}$: a set of shares, where $\mathcal{U} \coloneqq \bigcup_{i=1}^{n} \mathcal{U}_i$ and $\mathcal{U}_i \coloneqq \bigcup_{t=1}^{\tau} \mathcal{U}_i^{(t)}$;

$\mathcal{TI}$: a set of time-signals, where $\mathcal{TI} \coloneqq \bigcup_{t=1}^{\tau} \mathcal{TI}^{(t)}$.

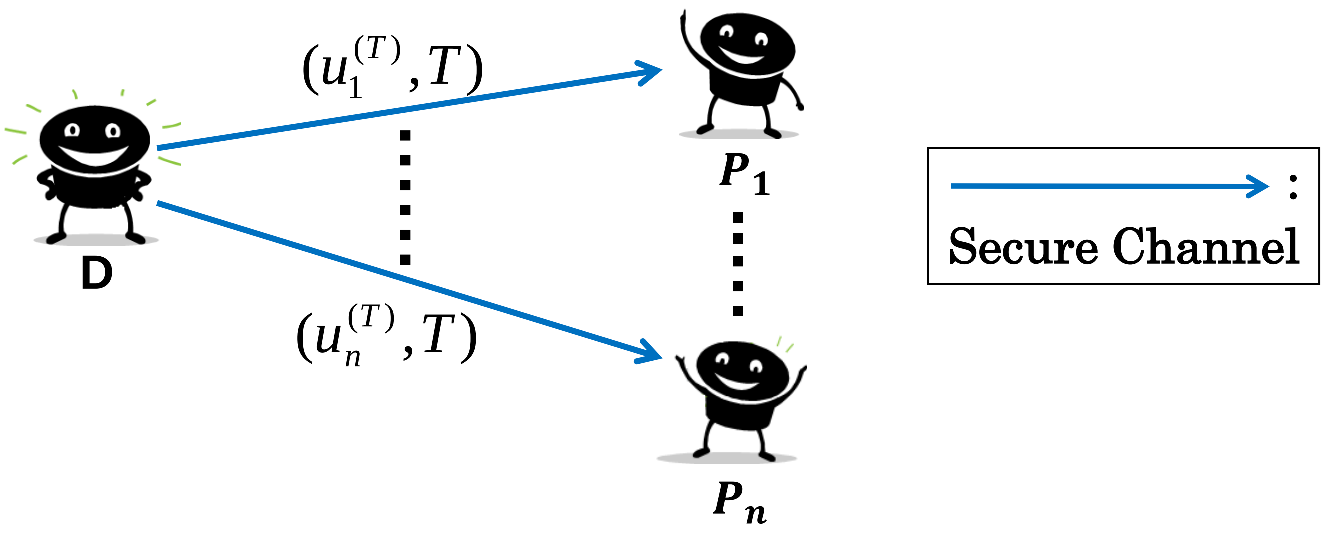
(k_1,k_2,n)-TR-SS: Model

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



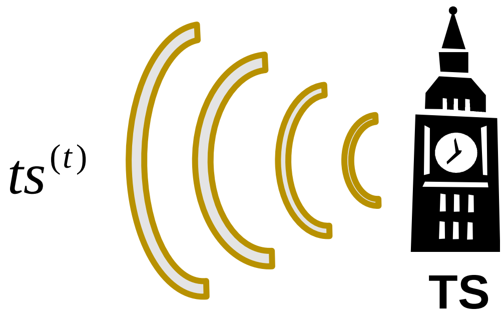
(k_1,k_2,n)-TR-SS: Model

2. **Share.**
   1. **D** randomly selects a secret $s \in \mathcal{S}$ and chooses k_1, k_2 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 $\left(u_i^{(T)}, T\right)$ to $\boldsymbol{P_i}$ via a secure channel $(i = 1,2,\ldots,n)$.



(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 $ts^{(t)} \in \mathcal{TI}^{(t)}$ by using his secret key sk.
    2. **TS** broadcasts $ts^{(t)}$.

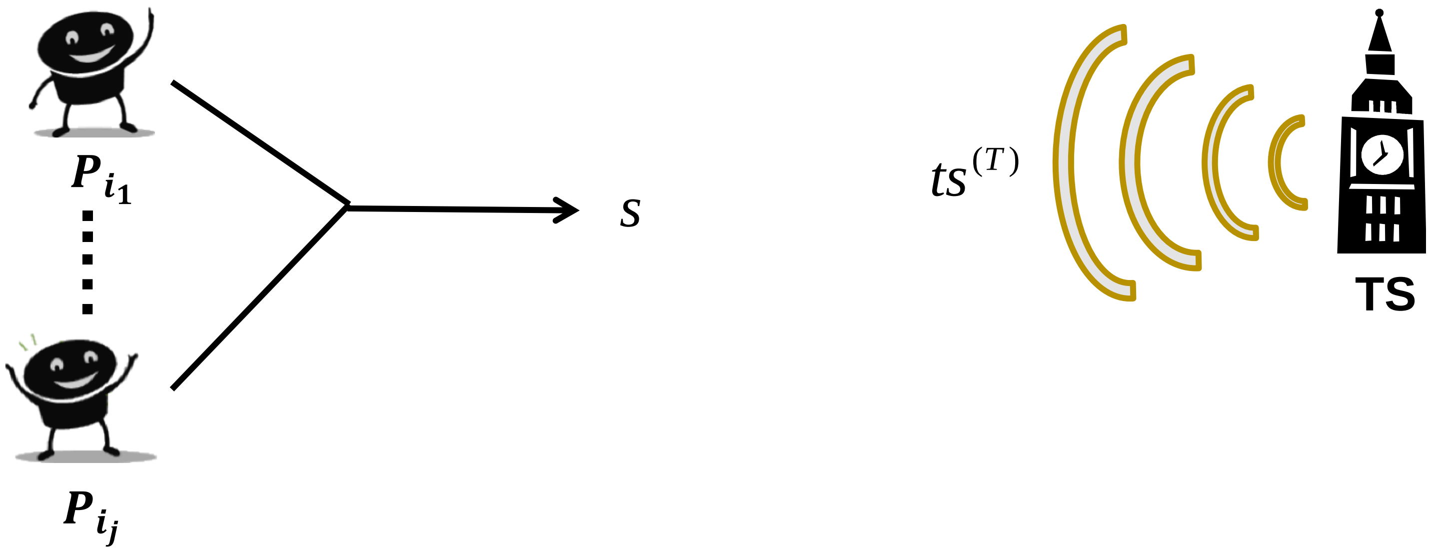


For simplicity, we assume $ts^{(t)}$ is deterministically computed by t and sk.

(k_1,k_2,n)-TR-SS: Model

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

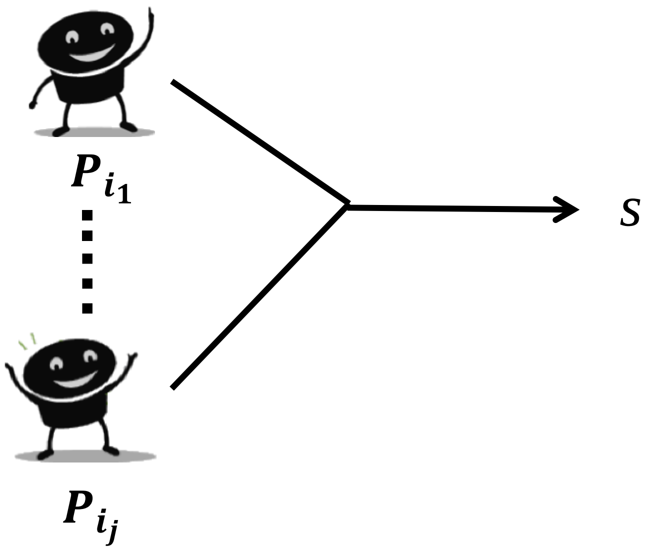
At the specified time T, any set of participants $A \coloneqq \{P_{i_1}, \dots, P_{i_j}\}$ $(k_1 \le j < k_2)$ can reconstruct s from their shares $u_{i_1}^{(T)}, \dots, u_{i_j}^{(T)}$ and a time-signal $ts^{(T)}$ at the specified time T.



(k_1,k_2,n)-TR-SS: Model

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

   At anytime, any set of participants $A \coloneqq \{P_{i_1}, \dots, P_{i_j}\}$ $(k_2 \le j \le n)$ can reconstruct s from **only** their shares $u_{i_1}^{(T)}, \dots, u_{i_j}^{(T)}$.



(k_1,k_2,n)-TR-SS: Security

We consider two kinds of security.

(i) Traditional secret sharing security.

(ii) Timed-release security.

Formally, a (k_1,k_2,n)-TR-SS scheme is *secure* if the following conditions are satisfied.

(i) For any $F \subset \mathcal{P}$ s.t. $1 \leq |F| \leq k_1 - 1$ and any $T \in \mathcal{T}$, it holds that

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

(ii) For any $\widehat{F} \subset \mathcal{P}$ s.t. $k_1 \leq \left|\widehat{F}\right| < k_2$ and any $T \in \mathcal{T}$, it holds that

$$H\left(S \mid U_{\widehat{F}}^{(T)}, TI^{(1)}, \ldots, TI^{(T-1)}, TI^{(T+1)}, \ldots, TI^{(\tau)} \right) = H(S).$$

(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.

<u>Theorem.</u>

For any $i \in \{1, 2, \dots, n\}$ and for any $T \in \mathcal{T}$, we have

$$\text{(i)} \quad H\left(U_i^{(T)}\right) \geq H(S).$$

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

$$\text{(ii)} \quad H\left(TI^{(T)}\right) \geq (k_2 - k_1)H(S),$$

$$\text{(iii)} \quad H(SK) \geq \tau(k_2 - k_1)H(S).$$

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.

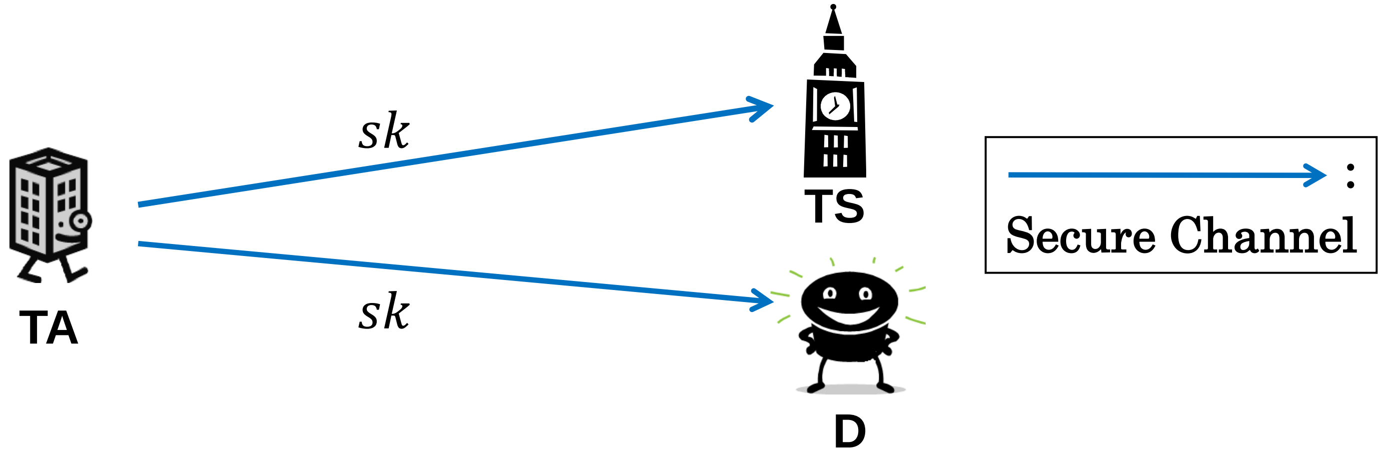
(k_1,k_2,n)-TR-SS: Optimal Construction

1. **Initialize.**

Let q be a prime power, where $q > \max(n, \tau)$.

Let $\mathbf{F}_q$ be a finite field with q elements.

1. **TA** chooses ℓ, which is the maximum difference between k_2 and k_1.
2. **TA** chooses $\ell \cdot \tau$ numbers $r_i^{(t)}$ $(i = 1, \dots, \ell,\ t = 1, \dots, \tau)$ from $\mathbf{F}_q$ uniformly at random.
3. **TA** sends $sk \coloneqq \left\{\left(r_1^{(t)}, \dots, r_\ell^{(t)}\right)\right\}_{1 \le t \le \tau}$ to **TS** and **D**, respectively.



(k_1, k_2, n)-TR-SS: Optimal Construction

1. **Initialize.**

   Let q be a prime power, where $q > \max(n, \tau)$.

   Let $\mathbf{F}_q$ be a finite field with q elements.

   1. **TA** chooses ℓ, which is the maximum difference between k_2 and k_1.
   2. **TA** chooses $\ell \cdot \tau$ numbers $r_i^{(t)}$ $(i = 1, \ldots, \ell,\ t = 1, \ldots, \tau)$ from $\mathbf{F}_q$ uniformly at random.
   3. **TA** sends $sk := \left\{\left(r_1^{(t)}, \ldots, r_\ell^{(t)}\right)\right\}_{1 \le t \le \tau}$ to **TS** and **D**, respectively.

Note.

This construction is optimal but **restricted**, since **D** will be only allowed to choose k_1 and k_2 s.t. $k_2 - k_1 \le \ell$ in the phase *Share*.

TA

D

(k_1,k_2,n)-TR-SS: Optimal Construction

2. **Share.**

   1. **D** randomly selects a secret $s \in \mathbf{F}_q$ 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 $\mathbf{F}_q$, where each a_i is chosen from $\mathbf{F}_q$ 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 $\left(u_i^{(T)}, T\right)$ to $\boldsymbol{P_i}$ 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 $s \in \mathbf{F}_q$ and chooses k_1, k_2 and n.
   2. **D** specifies future time $T \in \mathcal{T}$.
   3. **D** randomly chooses

Mask and disclose

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

over $\mathbf{F}_q$, where each a_i is chosen from $\mathbf{F}_q$ 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 $\left(u_i^{(T)}, T\right)$ to $\boldsymbol{P_i}$ 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

3. **Extract.**

At each time $t \in \mathcal{T}$**, TS** broadcasts t-th key $(r_1^{(t)}, \ldots, r_\ell^{(t)})$ as a time-signal at time t.

(k_1,k_2,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 $A := \left\{P_{i_1}, \ldots, P_{i_{k_1}}\right\}$ 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 $h\left(P_{i_j}\right) := f\left(P_{i_j}\right) - g\left(P_{i_j}\right)$ 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 $h(P_{i_1}), \ldots, h\left(P_{i_{k_1}}\right)$:
$$s = \sum_{j=1}^{k_1}\left(\prod_{l \neq j} \frac{P_{i_j}}{P_{i_j} - P_{i_l}}\right) h\left(P_{i_j}\right).$$

(k_1,k_2,n)-TR-SS: Optimal Construction

5. **Reconstruct without time-signals.**
   1. Any set of at least k_2 participants $\hat{A} := \{P_{i_1}, \ldots, P_{i_{k_2}}\}$ can compute s by Lagrange interpolation from $f(P_{i_1}), \ldots, f(P_{i_{k_2}})$:

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

Conclusion

- **Proposed Timed-Release Secret Sharing (TR-SS) schemes.**
  - 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.