Modeling Integrity

> *Created: 30 October 2025 | Last Updated: 30 October 2025*
>
>

Biba

中英雙語解釋 / Bilingual Explanation

為完整性建立標籤 / Establishing Labels for Integrity

就像 BLP 為保密性分配標籤一樣，我們可以為系統中的主體和客體關聯完整性標籤。
Suppose we associate integrity labels with subjects and with objects in our system, just as BLP did for confidentiality.

標籤的含義：

- **主體標籤：** 反映該主體的可信度。
  Subject Label: Should reflect the trustworthiness of the subject.

- **客體標籤：** 反映該客體中資訊的可靠性。
  Object Label: Should reflect the reliability of the information in the object.

重要說明： 完整性標籤與保密性權限標籤是分開的。
Important Proviso: Integrity labels are not the same as clearance labels.

一個同時強制執行保密性和完整性的系統，主體和客體必須擁有兩套獨立的標籤。
In a system that enforces both integrity and confidentiality, subjects/objects must have labels

for each.

例子： 一條資訊可能真實性可疑但非常敏感（低完整性，高保密性），也可能高度可靠但敏感性低（高完整性，低保密性）。
Example: A piece of information may be of dubious validity but very sensitive (Low Integrity, High Confidentiality), or highly reliable and of little sensitivity (High Integrity, Low Confidentiality).

完整性標籤的結構 / Structure of Integrity Labels

根據一個流行的模型（Biba 模型），完整性標籤的結構與 BLP 保密性標籤類似：
According to one popular model (the Biba model), integrity labels look like BLP confidentiality labels.

- **等級部分：** 表示可信度的等級（如：新手，學生，專家）。
Hierarchical Component: Gives the level of trustworthiness (e.g., Novice, Student, Expert).
- **範疇部分：** 提供相關能力領域的列表（如：物理，金融，醫學）。
Set of Categories: Provides a list of domains of relevant competence (e.g., Physics, Finance, Medicine).

例子： 一位物理學教授的完整性標籤可能是：(Expert: {Physics})
Example: A physics professor might have integrity label: (Expert: {Physics})

這意味著她在其專業領域內具有很高的可信度，但沒有理由相信她在政治或畜牧問題上的意見。
Meaning she has a high degree of credibility in her area of expertise, but there's no particular reason to trust her opinion on politics or animal husbandry.

支配關係同樣適用 / The Dominates Relation Applies

由於完整性標籤與 BLP 標籤結構相同，支配關係的定義也完全一樣。
Since integrity labels have the same structure as BLP labels, the dominates relation applies.

定義： (L1, C1) ≧ (L2, C2) 當且僅當：
Definition: (L1, C1) ≧ (L2, C2) iff:

- L1 ≥ L2 （在可信度等級上）
- C1 ⊇ C2 （在相關範疇上）

假設等級順序為： Novice < Student < Expert
Assume an ordered set of hierarchical levels: Novice < Student < Expert

- (Expert: {Physics}) 支配 (Student: {Physics})？ **是**
- (Student: {Physics}) 支配 (Expert: {Physics, Math})？ **否**（等級和範疇都不滿足）

完整性元策略 / The Integrity Metapolicy

現在是最關鍵的部分：完整性的規則應該是什麼？
But what are the rules? What is the metapolicy for integrity?

回顧一下，BLP 的元策略是約束資訊流以實現保密性："防止資訊從高保密級流向低保密級"。
Recall with MLS, the BLP rules were designed to constrain the flow of information for confidentiality: "prevent information from flowing from high to low confidentiality."

對於完整性，一個可能的元策略是："防止不可信的資訊'污染'可信的資訊。" 或者更簡單地說："不允許資訊在完整性上'向上流動'。"
For integrity, a possible metapolicy is: "Don't allow bad information to 'taint' good information."
Alternatively: "don't allow information to 'flow up' in integrity."

這意味著什麼？
一個低完整性（不可信）的資訊流入高完整性（可信）的客體，會污染後者，降低其可信度。這可以通過兩種方式發生：
What are the implications?
Low-integrity (untrustworthy) information flowing into a high-integrity (trustworthy) object would contaminate it. This can happen if:

- **"向上寫"**： 一個低完整性的主體將不可信的資訊寫入一個高完整性的客體。
"Write Up": A low-integrity subject writes untrustworthy information into a high-integrity object.

- **"向下讀"**： 一個高完整性的主體從低完整性的客體讀取了不可信的資訊，然後在處理過程中污染了它自己的高完整性資料。
"Read Down": A high-integrity subject reads bad information from a low-integrity object, and then, in its processing, contaminates its own high-integrity data.

因此，通過與 BLP 類比，可以推導出完整性規則：
This suggests, by analogy with the BLP rules, a subject shouldn't be allowed to:

- "Write Up" in integrity （禁止向上寫）
- "Read Down" in integrity （禁止向下讀）

這正好與 BLP 規則相反！BLP 是"無向上讀，無向下寫"，而完整性模型是"無向上寫，無向下讀"。

This is the opposite of BLP! BLP is "No Read Up, No Write Down". The integrity model is "No Write Up, No Read Down".

小測試 / Test

1. 一個主體的保密性標籤是 (Secret: {Crypto})，其完整性標籤是 (Unverified: {})。這描述了一個什麼樣的主體？
   A subject has a confidentiality label of (Secret: {Crypto}) and an integrity label of (Unverified: {}). What does this describe?

2. 根據推導出的完整性元策略，一個標籤為 (Novice: {}) 的使用者是否被允許向一個標籤為 (Expert: {Finance}) 的金融資料庫寫入資料？為什麼？
   According to the derived integrity metapolicy, is a user with the label (Novice: {}) allowed to WRITE to a financial database with the label (Expert: {Finance})? Why or why not?

3. 同樣，一個標籤為 (Expert: {Medicine}) 的醫療專家是否被允許從一個標籤為 (Novice: {}) 的公共維基網頁讀取資訊用於診斷？為什麼？
   Similarly, is a medical expert with the label (Expert: {Medicine}) allowed to READ from a public wiki page with the label (Novice: {}) for diagnostic purposes? Why or why not?

4. 保密性和完整性的核心控制目標有什麼根本不同？
   What is the fundamental difference in the core control objective between confidentiality and integrity?

答案與解析 / Answers and Explanations

1. **雙標籤主體**
   答案： 這描述了一個被高度信任接觸敏感密碼學資訊（高保密性），但其本身處理資訊的行為或產生的資訊卻不可靠（低完整性）的主體。例如，一個可以訪問秘密加密金鑰，但編寫程式碼很粗心、容易引入漏洞的程式。
   Answer: This describes a subject who is highly trusted to access sensitive cryptographic information (high confidentiality) but is itself unreliable in its actions or output (low integrity). For example, a program that has access to secret encryption keys but is carelessly written and prone to introducing vulnerabilities.

2. **"向上寫"規則**
   答案： 不允許。
   解析： 因為這將是"向上寫"。(Novice: {}) 的完整性等級遠低於 (Expert: {Finance})。允許新手向專家級資料庫寫入，會用不可信的資訊污染高度可信的資料源，違反"不允許資訊在完整性上向上流動"的元策略。

Answer: No.
Explanation: Because this would be a "Write Up". The integrity level of (Novice: {}) is much lower than (Expert: {Finance}). Allowing a novice to write to an expert-level database would contaminate a highly trustworthy data source with untrustworthy information, violating the "no flow up in integrity" metapolicy.

3. **"向下讀"規則**
答案： 不允許。
解析： 因為這將是"向下讀"。專家從新手級別的來源（如公共維基）讀取資訊，可能會在診斷中無意間使用這些不可靠的資訊，從而污染專家基於可靠醫學知識做出的判斷。這會降低診斷結果的完整性。
Answer: No.
Explanation: Because this would be a "Read Down". The expert reading from a novice-level source (like a public wiki) might inadvertently use this unreliable information in their diagnosis, contaminating their judgment based on reliable medical knowledge. This would lower the integrity of the diagnostic outcome.

4. **核心目標的根本區別**
答案： 保密性的核心是防止資訊洩露，關注的是資訊流向誰。完整性的核心是防止資訊被污染，關注的是資訊來自哪裡。因此，它們的訪問控制規則是鏡像相反的。BLP 保護"盒子"裡的內容不流出去；完整性模型保護"盒子"裡的內容不被從外面流入的髒東西污染。
Answer: Confidentiality's core is preventing disclosure, concerned with to whom information flows. Integrity's core is preventing contamination, concerned with from where information originates. Therefore, their access control rules are mirror opposites. BLP protects what's inside the "box" from flowing out; an integrity model protects what's inside the "box" from being contaminated by dirty things flowing in.

Biba Other Policies

中英雙語解釋 / Bilingual Explanation

比巴完整性模型 / Biba's Integrity Models

Ken Biba 在 1977 年提出了三種不同的完整性訪問控制策略：
Ken Biba proposed three different integrity access control policies:

1. 低水印完整性策略
The Low Water Mark Integrity Policy
2. 環策略

   The Ring Policy
3. 嚴格完整性
   Strict Integrity

其中只有嚴格完整性產生了持續的影響，現在通常被稱為比巴模型。
Only Strict Integrity had much continuing influence. It is the one typically referred to as the Biba Model or Biba Integrity.

嚴格完整性策略：BLP 的對偶 / The Strict Integrity Policy: The Dual of BLP

比巴嚴格完整性是一個強制性完整性訪問控制策略，並且是 BLP 的對偶。
The Strict Integrity Policy is a mandatory integrity access control policy and is the dual of BLP.

"對偶"是什麼意思？ 意味著比巴的規則與 BLP 的規則完全相反但結構對稱。
What does it mean to be the "dual" of BLP? It means Biba's rules are the exact opposite but structurally symmetrical to BLP's rules.

策略 / Policy	BLP (保密性)	Biba (完整性)
簡單屬性	無向上讀 Subject can read object only if $L_s \geq L_o$	無向下讀 Subject can read object only if $i_s \leq i_o$
*-屬性	無向下寫 Subject can write object only if $L_s \leq L_o$	無向上寫 Subject can write object only if $i_o \leq i_s$

比巴規則的具體表述：
Biba's Rules in Detail:

- **簡單完整性屬性：**
  主體 s 可以讀客體 o，僅當 $i_s \leq i_o$
  Simple Integrity Property:
  Subject s can read object o only if $i_s \leq i_o$
  解釋： 主體只能讀取與自己同級或更高完整性的客體。
  Interpretation: A subject can only read objects at its own integrity level or above.

- **完整性*-屬性：**
  主體 s 可以寫客體 o，僅當 $i_o \leq i_s$
  *Integrity -Property:
  Subject s can write to object o only if $i_o \leq i_s$

解釋： 主體只能寫入與自己同級或更低完整性的客體。
Interpretation: A subject can only write objects at its own integrity level or below.

規則解讀：如何保護完整性？ / Interpreting the Rules: How Do They Protect Integrity?

這些規則共同實現了完整性元策略：防止資訊在完整性上"向上流動"。
Together, these rules implement the integrity metapolicy: prevent information from "flowing up" in integrity.

簡單完整性屬性（無向下讀）的作用：
防止主體通過讀取低完整性（不可信）的資訊而被污染。
Simple Integrity (No Read Down) protects:
A subject's integrity cannot be tainted by reading bad (lower integrity) information.

例子： 一個醫療診斷程式（高完整性）不能從公共維基百科（低完整性）讀取醫療資訊，否則其診斷結果可能變得不可靠。
Example: A medical diagnosis program (high integrity) cannot read medical information from a public wiki (low integrity), or its diagnoses may become unreliable.

完整性*-屬性（無向上寫）的作用：
防止主體用低完整性的資料污染高完整性（更可靠）的資訊。
*Integrity -Property (No Write Up) protects:
A subject cannot taint more reliable (higher integrity) information by writing into it.

例子： 一個新手使用者（低完整性）不能向官方金融資料庫（高完整性）寫入資料，否則會污染資料庫的可靠性。
Example: A novice user (low integrity) cannot write data into an official financial database (high integrity), or they would contaminate the database's reliability.

嚴格完整性的訪問控制矩陣 / Strict Integrity Access Control Matrix

像所有訪問控制策略一樣，比巴模型可以用訪問控制矩陣表示。
Since this is an access control policy, it can be represented as an access control matrix.

假設完整性等級：H (高) > L (低)
Assume integrity levels: H (High) > L (Low)

Subject \ Object	Obj_1 (H)	Obj_2 (L)
$Subj_1$ (H)	R, W	W
$Subj_2$ (L)	R	R, W

驗證：

- Subj_1 (H) → Obj_1 (H): 讀？ H ≤ H ✓ 寫？ H ≤ H ✓ → R, W
- Subj_1 (H) → Obj_2 (L): 讀？ H ≤ L ✕ 寫？ L ≤ H ✓ → W
- Subj_2 (L) → Obj_1 (H): 讀？ L ≤ H ✓ 寫？ H ≤ L ✕ → R
- Subj_2 (L) → Obj_2 (L): 讀？ L ≤ L ✓ 寫？ L ≤ L ✓ → R, W

這個矩陣與 BLP 的矩陣完全不同，體現了相反的控制目標。

結合 BLP 和比巴完整性 / Combining BLP and Biba Integrity

由於保密性和完整性是正交的（相互獨立的），要同時保護兩者，可以同時使用 BLP 和比巴策略。
Since confidentiality and integrity are orthogonal, to protect both, one could use both BLP and Biba's Strict Integrity policy.

實現方式：

- 所有主體和客體都需要兩套標籤：一套保密性標籤，一套完整性標籤。
  You'd need confidentiality labels and integrity labels for all subjects and objects.
- 一次訪問請求必須同時滿足 BLP 規則和比巴規則才會被允許。
  An access is allowed only if allowed by both the BLP rules and the Biba rules.

示例： 一個主體請求讀一個客體。
Example: A subject requests READ access to an object.

- BLP 檢查： L_s(保密性) ≥ L_o(保密性)？ （必須為真）
- Biba 檢查： i_s(完整性) ≤ i_o(完整性)？ （必須為真）

只有兩個條件都滿足，讀操作才被允許。

小測試 / Test

1. 一個主體的完整性標籤是 H，一個客體的完整性標籤是 L。根據比巴模型，該主體能否寫入該客體？為什麼？
   A subject has integrity label H, an object has integrity label L. According to the Biba model, can the subject WRITE to the object? Why?

2. 同樣的情況下，該主體能否讀取該客體？為什麼？
   In the same situation, can the subject READ from the object? Why?

3. 如果一個系統同時執行 BLP（保密性）和比巴（完整性），一個訪問請求需要滿足什麼條件才會被允許？
   If a system enforces both BLP (confidentiality) and Biba (integrity), what condition must an access request satisfy to be allowed?

4. 比巴模型的"無向上寫"規則主要防止什麼風險？
   What risk does the Biba model's "No Write Up" rule primarily prevent?

答案與解析 / Answers and Explanations

1. **寫入低完整性客體**
   答案： 可以
   解析： 根據比巴的完整性*-屬性，寫入操作要求 $i_o \leq i_s$。這裡 $L \leq H$ 成立。這符合"無向上寫"原則：允許高完整性主體向低完整性客體寫入，因為這會用可靠資訊覆蓋不可靠資訊，不會污染高完整性源。
   Answer: Yes
   Explanation: According to Biba's Integrity *-Property, write access requires $i_o \leq i_s$. Here, $L \leq H$ is true. This aligns with "No Write Up": allowing a high-integrity subject to write to a low-integrity object is safe, as it overwrites unreliable information with reliable information, without contaminating high-integrity sources.

2. **讀取低完整性客體**
   答案： 不可以
   解析： 根據比巴的簡單完整性屬性，讀取操作要求 $i_s \leq i_o$。這裡 $H \leq L$ 不成立。這執行了"無向下讀"原則：防止高完整性主體被低完整性客體內的不信任資訊所污染。
   Answer: No
   Explanation: According to Biba's Simple Integrity Property, read access requires $i_s \leq i_o$. Here, $H \leq L$ is false. This enforces "No Read Down": it prevents a high-integrity subject from being contaminated by untrustworthy information in a low-integrity object.

3. **結合執行的條件**
   答案： 該訪問請求必須同時滿足 BLP 的所有規則和比巴的所有規則。例如，一個讀操作必須同時通過 BLP 的"簡單安全屬性"（無向上讀）和比巴的"簡單完整性屬性"（無向下讀）的檢查。
   Answer: The access request must be allowed by all rules of BLP AND all rules of Biba. For example, a read operation must pass both BLP's Simple Security Property (No Read Up) and Biba's Simple Integrity Property (No Read Down).

4. **"無向上寫"防止的風險**
   答案： 主要防止不可信（低完整性）的資訊污染或破壞可信（高完整性）的資訊源。例

如，防止一個普通使用者修改關鍵的系統配置檔案，或者防止一個未經審查的程式寫入重要的資料庫。
Answer: It primarily prevents untrustworthy (low-integrity) information from contaminating or corrupting trustworthy (high-integrity) information sources. For example, preventing a regular user from modifying critical system configuration files, or preventing an unvetted program from writing to an important database.

Lipner Model

中英雙語解釋 / Bilingual Explanation

商業完整性需求 / Commercial Integrity Constraints

Steve Lipner (來自 Microsoft) 確定了商業資料處理環境中的特定完整性問題：
Steve Lipner describes integrity concerns you might find in a commercial data processing environment:

- 使用者不應自己編寫程式，而應使用現有的生產軟體。
  Users will not write their own programs, but use existing production software.
- 程式設計師在非生產系統上使用虛構資料開發和測試應用程式。
  Programmers develop and test applications on a nonproduction system, possibly using contrived data.
- 將應用程式從開發環境移動到生產環境需要一個受控和審計的特殊流程。
  Moving applications from development to production requires a special process. This process must be controlled and audited.
- 管理者和審計員必須能夠訪問系統狀態和系統日誌。
  Managers and auditors must have access to system state and system logs.

問題： 我們能使用現有的建模機制（BLP 和 Biba）來構建滿足這些約束的安全系統嗎？
Can we use our existing modeling mechanisms to build a secure system that addresses such constraints?

Lipner 的完整性矩陣模型 / Lipner's Integrity Matrix Model

Lipner 設計了他的完整性矩陣模型，通過結合 BLP 和 Biba 完整性來處理這些商業需求。
Lipner devised his Integrity Matrix Model to handle those concerns via a combination of BLP and Biba Integrity.

模型結構：

The Model Structure:

A. 保密性標籤 / Confidentiality Labels:

- 2 個等級：
  - 審計管理員： 用於系統審計和管理。
    Audit Manager (AM): system audit and management.
  - 系統低點： 所有其他行程。
    System Low (SL): all other processes.
- 3 個範疇：
  - 生產： 生產程式碼和資料。
    Production (SP): production code and data.
  - 開發： 開發中的程式。
    Development (SD): programs under development.
  - 系統開發： 開發中的系統程式。
    System Development (SSD): system programs in development.

B. 完整性標籤 / Integrity Labels:

- 3 個等級（從高到低）：
  - 系統程式： 系統軟體。
    System Program (ISP): system software.
  - 操作級： 生產程式和開發軟體。
    Operational (IO): production programs and development software.
  - 系統低點： 使用者級行為。
    System Low (ISL): user level behavior.
- 2 個範疇：
  - 開發： 開發實體。
    Development (ID): development entities.
  - 生產： 生產實體。
    Production (IP): production entities.

標籤分配 / Label Assignments

主體標籤分配： 基於組織中的角色和"須知"需求分配安全等級（保密性和完整性）。
Security levels are assigned to subjects based on their roles in the organization and their need to know.

示例主體：

- 普通使用者： (SL, {SP}) / (ISL, {IP})
- 程式設計師： (SL, {SD, SSD}) / (IO, {ID})
- 系統管理員： (AM, {SP, SD, SSD}) / (ISP, {IP, ID})

客體標籤分配： 基於誰應該訪問它們來分配安全等級。
Security levels are assigned to objects based on who should access them.

示例客體：

- 生產資料： (SL, {SP}) / (IO, {IP})
- 開發中程式碼： (SL, {SD}) / (IO, {ID})
- 系統日誌： (AM, {SP, SD, SSD}) / (ISP, {IP, ID})

關鍵設計思想： 普通使用者需要修改生產資料但不能修改生產程式碼。
Ordinary users need to modify production data but not production code.

完整性模型允許這種區別，而保密性模型為此移除了某些寫約束。
The integrity model allows this; note that writing constraints are removed from the confidentiality model for this purpose.

模型應用示例 / Applying the Model

讓我們用這個模型回答一些關鍵問題：
Let's use this model to answer some key questions:

1. **普通使用者能否使用系統程式？能否修改它？**

    i. Can an ordinary user utilize a system program? Modify it?
    - 普通使用者標籤： (SL, {SP}) / (ISL, {IP})
    - 系統程式標籤： 假設為 (SL, {SP}) / (ISP, {IP})
    - 使用（讀操作）：
        - BLP 檢查： 使用者等級 SL ≥ 程式等級 SL ✓，使用者範疇 {SP} ⊇ 程式範疇 {SP} ✓ → 允許讀
        - Biba 檢查： 使用者完整性 ISL ≤ 程式完整性 ISP ✓ → 允許讀
        - **結論：可以讀取/使用**
    - 修改（寫操作）：
        - Biba 檢查： 程式完整性 ISP ≤ 使用者完整性 ISL？ ✕ （ISP > ISL） → 禁止寫
        - **結論：不能修改**

2. **系統程式設計師能否使用生產軟體？能否修改它？**

   ii. Can a system programmer use production software? Modify it?

   - 程式設計師標籤： (SL, {SD, SSD}) / (IO, {ID})
   - 生產軟體標籤： (SL, {SP}) / (IO, {IP})
   - 使用（讀操作）：
     - BLP 檢查： 程式設計師範疇 {SD, SSD} ⊇ 軟體範疇 {SP}？ ✕ → 禁止讀
     - **結論：既不能讀取也不能修改（因為讀都不允許）**

3. **為什麼需要特殊的降級權限？僅用 BLP 和 Biba 能實現嗎？**

   iii. Why is that special downgrade permission required? Could it be done with BLP and Biba alone?

   - **需要降級的原因：** 將物件從開發環境移動到生產環境意味著改變它們的標籤（從開發標籤變為生產標籤）。
   - **BLP/Biba 的局限性：** 在標準的 BLP 和 Biba 中，標籤是強制性的且遵循 ** tranquility 屬性**，不允許隨意更改。沒有明顯的方法可以在這些模型內完成這種有控制的標籤變更。
   - **Lipner 的解決方案：** 需要修改 tranquility 屬性，建立一個特殊的、受控的流程來授權這種標籤更改（降級）。

小測試 / Test

1. 在 Lipner 的模型中，一個標籤為 (SL, {SP}) / (ISL, {IP}) 的普通使用者想要修改一個標籤為 (SL, {SP}) / (IO, {IP}) 的生產資料檔案。根據 Biba 的完整性規則，這個寫操作會被允許嗎？為什麼？
   In Lipner's model, an ordinary user with label (SL, {SP}) / (ISL, {IP}) wants to MODIFY a production data file with label (SL, {SP}) / (IO, {IP}). According to Biba's integrity rules, would this write operation be allowed? Why?

2. 同一個使用者想要讀取系統審計日誌（假設標籤為 (AM, {SP}) / (ISP, {IP})）。根據 BLP 的保密性規則，這個讀操作會被允許嗎？為什麼？
   The same user wants to READ the system audit logs (assume label (AM, {SP}) / (ISP, {IP})). According to BLP's confidentiality rules, would this read operation be allowed? Why?

3. Lipner 模型中的"降級"操作違反了 BLP/Biba 的哪個核心屬性？為什麼這種違反在商業環境中是必要的？
   The "downgrade" operation in Lipner's model violates which core property of BLP/Biba? Why is this violation necessary in a commercial environment?

4. Lipner 模型是更偏向"強制訪問控制"還是"自主訪問控制"？為什麼？

Is Lipner's model more aligned with "Mandatory Access Control" or "Discretionary Access Control"? Why?

答案與解析 / Answers and Explanations

1. **使用者修改生產資料**
   答案： 允許
   解析： 根據 Biba 的完整性*-屬性，寫操作要求 $i_o \le i_s$。這裡，資料檔案的完整性等級 IO ≤ 使用者的完整性等級 ISL？不，IO > ISL（操作級 > 系統低級）。因此，寫操作會被拒絕。這正好實現了商業需求：防止低完整性的使用者直接修改高完整性的生產資料。但請注意，Lipner 可能在實際規則上做了調整來允許這種必要的業務操作。
   Answer: No
   Explanation: According to Biba's Integrity *-Property, write requires $i_o \le i_s$. Here, the data file's integrity level IO ≤ user's integrity level ISL? No, IO > ISL (Operational > System Low). Therefore, the write operation would be denied. This actually implements the commercial requirement: preventing low-integrity users from directly modifying high-integrity production data. (Note: Lipner might have adjusted the actual rules to allow this necessary business operation.)

2. **使用者讀取審計日誌**
   答案： 不允許
   解析： 根據 BLP 的簡單安全屬性，讀操作要求 $L_s \ge L_o$。使用者的保密性等級 SL ≥ 日誌的保密性等級 AM？不，SL < AM（系統低級 < 審計管理員）。因此，讀操作會被拒絕。這實現了審計日誌的保密性：只有審計管理員可以讀取。
   Answer: No
   Explanation: According to BLP's Simple Security Property, read requires $L_s \ge L_o$. The user's confidentiality level SL ≥ log's confidentiality level AM? No, SL < AM (System Low < Audit Manager). Therefore, the read operation would be denied. This implements confidentiality for audit logs: only audit managers can read them.

3. **降級操作的必要性**
   答案： 它違反了 tranquility 屬性（特別是強 tranquility 屬性），該屬性規定標籤在系統生命週期內不應改變。
   解析： 這種違反在商業環境中是必要的，因為軟體開發生命週期要求將程式碼從"開發"狀態（低可信度）提升到"生產"狀態（高可信度）。這是一個受控的、經過授權的流程，而不是任意的更改。標準的 BLP/Biba 沒有為這種必要的狀態轉換提供機制。
   Answer: It violates the Tranquility Property (specifically the Strong Tranquility Property), which states that labels should not change during the system's lifetime.
   Explanation: This violation is necessary in a commercial environment because the

software development lifecycle requires promoting code from a "development" state (low trust) to a "production" state (high trust). This is a controlled, authorized process, not an arbitrary change. Standard BLP/Biba provides no mechanism for this necessary state transition.

4. **模型的訪問控制類型**
   答案： 更偏向強制訪問控制。
   解析： 因為訪問決策基於系統強制執行的、與使用者角色和資料類型綁定的安全標籤（保密性和完整性）。普通使用者不能隨意更改自己或資料的標籤。這與基於所有者的自主訪問控制（如 Unix 檔案權限）相反。
   Answer: It is more aligned with Mandatory Access Control.
   Explanation: Because access decisions are based on system-enforced security labels (confidentiality and integrity) tied to user roles and data types. An ordinary user cannot arbitrarily change their own or the data's labels. This contrasts with owner-based Discretionary Access Control (like Unix file permissions).

Clark-Wilson Model

中英雙語解釋 / Bilingual Explanation

Clark-Wilson 模型的背景 / Background of the Clark-Wilson Model

Lipner 模型的局限性： Lipner 的完整性矩陣模型表明 BLP 和 Biba 可以適配成可用的商業策略，但不一定是最佳方案。
Limitation of Lipner's Model: Lipner's model showed BLP and Biba can be adapted to yield a workable commercial policy, but not necessarily a good fit.

Clark-Wilson 的論點： David Clark 和 David Wilson 認為商業安全有自己獨特的需求，值得為之專門設計一個模型。
Clark-Wilson's Argument: Commercial security has its own unique concerns and merits a model crafted for that domain.

核心關注點： 系統狀態各個組件之間的一致性。
Overriding Concern: Consistency among the various components of the system state.

完整性的定義： 完整性由一組約束條件來定義。當資料滿足這些約束時，它就處於一致或有效的狀態。
Definition of Integrity: Integrity is defined by a set of constraints. Data is in a consistent or valid state when it satisfies these constraints.

銀行示例 / Bank Example

在銀行中，期初資金 X + 存入資金 Y - 取出資金 Z = 期末資金 W。
In a bank: Funds X at beginning of day + deposits Y - withdrawals Z = funds on hand W at end of day.

這個例子提出了完整性約束：X + Y - Z = W
This suggests integrity constraints to ensure: X + Y - Z = W

良構事務： 必須將系統從一個一致狀態轉移到另一個一致狀態。
Well-formed transactions must move system from one consistent state to another.

關鍵問題： 誰檢查、認證事務被正確執行？
Issue: Who examines, certifies transactions done correctly?

商業完整性的四個基本關注點 / Four Fundamental Concerns of Commercial Integrity

Clark 和 Wilson 聲稱，任何合理的商業完整性模型都必須關注以下四點：
Clark and Wilson claimed that the following are four fundamental concerns of any reasonable commercial integrity model:

1. **認證：** 所有使用者的身份必須被正確認證。
   Authentication: Identity of all users must be properly authenticated.
2. **審計：** 修改操作應該被記錄，以不可篡改的方式記錄每個程式由誰執行。
   Audit: Modifications should be logged to record every program executed and by whom, in a way that cannot be subverted.
3. **良構事務：** 使用者只能以受約束的方式操作資料。只允許合法的訪問。
   Well-formed transactions: Users manipulate data only in constrained ways. Only legitimate accesses are allowed.
4. **職責分離：** 系統將每個使用者與一組他們可以運行的有效程式關聯起來，並防止未授權的修改，從而保持與現實世界的一致性和完整性。
   Separation of duty: The system associates with each user a valid set of programs they can run and prevents unauthorized modifications, thus preserving integrity and consistency with the real world.

核心概念 / Key Concepts

該策略基於以下類別構建：
The policy is constructed in terms of the following categories:

概念 / Concept	描述 / Description

受約束資料項	其完整性受保護的物件（如銀行帳戶餘額）。
Constrained Data Items (CDIs)	Objects whose integrity is protected (e.g., bank account balances).
無約束資料項	完整性策略不覆蓋的物件（如使用者輸入的資料）。
Unconstrained Data Items (UDIs)	Objects not covered by the integrity policy (e.g., user input data).
轉換過程	唯一被允許修改 CDIs 的過程，或將任意使用者輸入建立為新 CDIs 的過程。
Transformation Procedures (TPs)	The only procedures allowed to modify CDIs, or take user input and create new CDIs.
完整性驗證過程	用於驗證 CDIs 完整性是否得到維護的過程。
Integrity Verification	Procedures meant to verify maintenance of integrity of CDIs.
Procedures (IVPs)	

TPs 的設計目的： 將系統從一個有效狀態轉移到另一個有效狀態。
TPs are designed to take the system from one valid state to another.

策略規則：認證規則與執行規則 / Policy Rules: Certification vs. Enforcement

Clark-Wilson 模型有兩類規則：

認證規則： 規定系統應該如何被設計和驗證。
Certification Rules: How the system should be designed and verified.

- **C1:** 所有 IVPs 必須確保當 IVP 運行時，CDIs 處於有效狀態。
  C1: All IVPs must ensure that CDIs are in a valid state when the IVP is run.
- **C2:** 所有 TPs 必須被認證為能保持完整性。
  C2: All TPs must be certified as integrity-preserving.
- **C3:** TPs 分配給使用者必須滿足職責分離。
  C3: Assignment of TPs to users must satisfy separation of duty.
- **C4:** TPs 的操作必須被記錄。
  C4: The operation of TPs must be logged.
- **C5:** 在 UDIs 上執行的 TPs 必須產生有效的 CDIs。
  C5: TPs executing on UDIs must result in valid CDIs.

執行規則： 規定系統在運行時必須強制執行什麼。
Enforcement Rules: What the system must enforce at runtime.

- **E1:** 只有被認證的 TPs 才能操作 CDIs。
  E1: Only certified TPs can manipulate CDIs.
- **E2:** 使用者只能通過他們被授權的 TPs 來訪問 CDIs。
  E2: Users must only access CDIs by means of TPs for which they are authorized.
- **E3:** 必須認證每個嘗試執行 TP 的使用者的身份。
  E3: The identity of each user attempting to execute a TP must be authenticated.
- **E4:** 只有 TP 的認證者才能更改與該 TP 關聯的實體列表。
  E4: Only the certifier of a TP may change the list of entities associated with that TP.

關鍵規則詳解 / Elaborations on Key Rules

C2 和 E1/E2（核心訪問控制）：

- 定義一個認證關係，將一組 CDIs 與特定的 TP 關聯起來。
  Defines a certified relation that associates a set of CDIs with a particular TP.
- **例子：** 在銀行中，餘額查詢 TP 被認證可以操作 帳戶集合 CDIs。
  Example: TP balance, CDIs accounts in the bank example.
- 系統維護這個關係，並確保只有被認證可以操作某個 CDI 的 TP 才能操作它。

C3 和 E3（職責分離和認證）：

- 允許關係（將 TPs 分配給使用者）必須滿足職責分離要求。
  Allowed relations (assignment of TPs to users) must meet separation of duty.
- **例子：** 建立採購訂單的 TP 和授權付款的 TP 必須分配給不同的使用者。
  Example: The TP to create a purchase order and the TP to authorize payment must be assigned to different users.

C4（審計）：

- 所有 TPs 必須將足夠重建操作的資訊追加到一個只追加的 CDI（即審計日誌）中。
  All TPs must append enough information to an append-only CDI (the log).

C5（處理不受信任的輸入）：

- 任何以 UDI 為輸入的 TP，必須對所有可能的 UDI 值執行有效轉換或不執行轉換。
  Any TP that takes a UDI may perform only valid transformations, or no transformations, for all possible values of the UDI.
- **例子：** 從鍵盤輸入的數字是 UDIs。TPs 必須先驗證這些數字（使它們成為 CDI）才能使用它們。
  Example: Numbers entered at keyboard are UDIs. TPs must validate numbers before using

them.

與 Biba 模型的比較 / Comparison to Biba

特性 / Aspect	Biba 模型	Clark-Wilson 模型
方法	基於標籤的強制訪問控制	基於程式/事務的訪問控制
信任	信任主體會遵守規則	信任認證過程和事務邏輯
資料處理	在分配標籤前檢查不受信任的資料	信任實體必須認證升級方法（而不是資料本身）
核心機制	安全標籤和支配關係	認證關係、允許關係、轉換過程

訪問控制實現 / Access Control Implementation

在 Clark-Wilson 中，權限被編碼為形式為的三元組集合：
Allowed relations or permissions are encoded as a set of triples:

```
(user, TP, {CDI set})
```

含義： 使用者被授權在給定的 CDI 集合上執行事務過程 TP。
Meaning: user is authorized to perform TP on the given set of CDIs.

例子： (Alice, Process_Payment, {Vendor_Accounts})
這意味著 Alice 被授權在供應商帳戶集合上執行"處理付款"事務。

每個這樣的三元組必須符合所有適用的認證和執行規則。

小測試 / Test

1. 在 Clark-Wilson 模型中，轉換過程 和 完整性驗證過程 的主要區別是什麼？
   What is the main difference between a Transformation Procedure (TP) and an Integrity Verification Procedure (IVP) in the Clark-Wilson model?

2. 根據規則 E2，使用者 Alice 能否直接修改一個受約束資料項（CDI），比如直接更改資料庫中的銀行帳戶餘額？為什麼？
   According to rule E2, can user Alice directly modify a Constrained Data Item (CDI), such as directly changing a bank account balance in the database? Why or why not?

3. 規則 C5 主要解決什麼安全問題？請用銀行例子說明。
   What security problem does rule C5 primarily address? Illustrate with a bank example.

4. Clark-Wilson 模型中的三元組 (Bob, Approve_Loan, {Customer_Loan_Applications}) 表示什麼？
   What does the triple (Bob, Approve_Loan, {Customer_Loan_Applications}) represent in the Clark-Wilson model?

答案與解析 / Answers and Explanations

1. **TP vs IVP**
   答案： TP 是修改 CDIs 的過程，負責將系統從一個有效狀態轉換到另一個有效狀態。IVP 是驗證 CDIs 是否處於有效狀態的過程，但不修改它們。TPs 用於日常操作，IVPs 用於定期審計或系統啟動時檢查。
   Answer: A TP is a procedure that modifies CDIs, moving the system from one valid state to another. An IVP is a procedure that verifies CDIs are in a valid state, but does not modify them. TPs are for daily operations; IVPs are for periodic audits or startup checks.

2. **直接修改 CDI**
   答案： 不能。
   解析： 規則 E2 明確規定"使用者必須只能通過他們被授權的 TPs 來訪問 CDIs"。直接修改繞過了一致性檢查和審計日誌，違反了良構事務的原則。Alice 必須使用一個被認證的"更新餘額" TP，這個 TP 會執行必要的檢查（如防止餘額為負）並記錄日誌。
   Answer: No.
   Explanation: Rule E2 explicitly states "Users must only access CDIs by means of TPs for which they are authorized." Direct modification bypasses consistency checks and audit logging, violating the principle of well-formed transactions. Alice must use a certified "update-balance" TP that performs necessary checks (e.g., preventing negative balance) and logs the action.

3. **規則 C5 的目的**
   答案： 規則 C5 主要解決處理不受信任輸入的安全問題。它確保來自外部世界（使用者輸入、網路資料）的資料在被系統信任和使用之前必須經過嚴格的驗證和清理。
   銀行例子： 當櫃員在終端輸入存款金額"100.00"時，這個輸入是一個 UDI。存款 TP（一個 TP）必須驗證這個輸入：它是正數嗎？格式正確嗎？只有通過驗證後，這個數值才能被用來更新帳戶餘額（一個 CDI）。如果輸入是"-$100"或"ABC"，TP 應該拒絕它。
   Answer: Rule C5 primarily addresses the problem of handling untrusted input. It ensures that data from the outside world (user input, network data) must be rigorously validated and sanitized before being trusted and used by the system.
   Bank Example: When a teller types a deposit amount "100.00" into a terminal, this input is a UDI. The deposit TP must validate this input: Is it a positive number? Is the format correct? Only after validation can this value be used to update the account balance (a

CDI). If the input were "-$100" or "ABC", the TP should reject it.

4. **三元組的含義**
   答案： 這個三元組是允許關係的一部分。它表示使用者"Bob"被授權在"Customer_Loan_Applications"這個受約束資料項集合上執行"Approve_Loan"這個轉換過程。Bob 不能批准其他類型的貸款申請，也不能執行其他與貸款相關的操作（如建立申請），除非有其他三元組授權他這樣做。
   Answer: This triple is part of the allowed relation. It represents that user "Bob" is authorized to execute the "Approve_Loan" Transformation Procedure on the set of CDIs called "Customer_Loan_Applications". Bob cannot approve other types of loan applications, nor can he perform other loan-related actions (like creating applications), unless other triples authorize him to do so.

Chinese Wall Policy

中英雙語解釋 / Bilingual Explanation

中國牆策略的背景 / Background of the Chinese Wall Policy

之前的策略都比較通用，而中國牆策略針對一個非常具體的商業問題：
The policies so far have been general. Let's consider a policy for a very specific commercial concern:

利益衝突： 顧問或承包商可能存在的利益衝突和無意間洩露資訊的問題。
Conflicts of interest and inadvertent disclosure of information by a consultant or contractor.

例子： 一個專門處理產品責任案件的律師為美國航空公司提供諮詢。如果她也為聯合航空公司提供諮詢，就可能違反保密原則。
Example: A lawyer specializing in product liability consults for American Airlines. It could be a breach of confidentiality for her to consult also for United Airlines.

為什麼是衝突？ 因為她給任何一方的建議都會影響她給另一方的建議。
Why? Her advice for either airline would affect her advice to the other airline.

為什麼不是衝突？ 同時為麥當勞提供服務就不會有衝突（不同行業）。
Why not? A simultaneous contract with McDonalds would not be a conflict.

策略的核心概念 / Core Concepts of the Policy

Brew 和 Nash（1989）提出了這個策略。嚴格來說，這不是完整性策略，而是訪問控制保密性

策略。
Brewer and Nash proposed this to address such conflicts of interest. Strictly speaking, this is not an integrity policy, but an access control confidentiality policy.

策略建立在三個抽象層次上：
The security policy builds on three levels of abstraction:

1. **物件：** 包含關於一家公司資訊的檔案。
   Objects: such as files containing information about only one company.
2. **公司資料集：** 包含特定公司所有物件的集合。表示為 CD(o)。
   Company Dataset (CD): contains all objects concerning a particular company. Denoted as CD(o).
3. **利益衝突類：** 包含相互競爭公司的資料集。假設每個物件只屬於一個 COI 類。
   Conflict of Interest Classes (COI): contain datasets of companies in competition. Assume each object belongs to exactly one COI class.

COI 類示例：

- {福特, 克萊斯勒, 通用汽車}（汽車製造商）
- {匯豐銀行, 渣打銀行, 花旗銀行}（銀行）
- {微軟}（單獨一類，因為沒有直接競爭對手在集合中）

簡單訪問控制規則 / The Simple Access Control Rule

核心規則： 一個主體可以訪問任何公司的資訊，只要該主體從未訪問過同一利益衝突類中不同公司的資訊。
A subject may access information from any company as long as that subject has never accessed information from a different company in the same conflict class.

例子： 如果你訪問了通用汽車的檔案，隨後你將被阻止訪問福特或克萊斯勒的任何檔案。但你仍然可以自由訪問任何其他利益衝突類中的公司檔案。
Example: If you access a file from GM, you subsequently will be blocked from accessing any files from Ford or Chrysler. You are free to access files from companies in any other conflict class.

關鍵特性：權限動態變化！
Notice that permissions change dynamically.

主體享有的訪問權限取決於過去的訪問歷史。
The access rights that any subject enjoys depends on the history of past accesses.

淨化資料 / Sanitization

公共資訊可能屬於公司資料集，但由於是公開可用的，不會引起利益衝突。
Public information may belong to a company dataset. As it is publicly available, no conflicts of interest arise.

因此，它不應影響分析師讀取其他資訊的能力。
Thus, it should not affect ability of analysts to read.

通常，在公開發布之前，所有敏感資料都從這類資訊中移除（稱為淨化）。
Typically, all sensitive data is removed from such information before it is released publicly (called sanitization).

正式策略規則 / Formal Policy Rules

策略根據以下兩個屬性限制訪問：
The policy restricts access according to the following two properties:

1. **中國牆簡單安全規則（管理讀訪問）：**

    i. (Chinese Wall) Simple Security Rule (manages read access):

    主體 s 可以被授予對物件 o 的訪問權，僅當該物件：
    A subject s can be granted access to object o only if the object:

    - 在 s 已經訪問過的物件所在的同一公司資料集內（即"在牆內"），或
      is in the same company datasets as the objects already accessed by s (i.e., "within the Wall"), or
    - 屬於完全不同的利益衝突類，或
      belongs to an entirely different conflict of interest class, or
    - 是淨化物件。
      is a sanitized object.

2. **中國牆*-屬性（管理寫訪問）：**
    *2. (Chinese Wall) -Property (manages write access):

    主體 s 可以寫入物件 o，當且僅當以下兩個條件都成立：
    A subject s can write to o iff both of the following hold:

    - CW 簡單安全規則允許 s 讀取 o；且
      The CW-simple security rule permits s to read o; and
    - 對於所有未淨化物件 o'，如果 s 可以讀取 o'，則 CD(o') = CD(o)。

For all unsanitized objects o', if s can read o', then CD(o′) = CD(o).

-屬性的含義： s 可以寫入一個物件，僅當它能讀取的所有（未淨化的）物件都在同一個公司資料集中。這避免了間接的利益衝突。
It says that s can write to an object if all the (unsanitized) objects it can read are in the same company dataset. This avoids indirect conflicts of interest.

寫訪問的例子 / Example of Write Access Control

假設 Anthony 和 Susan 在同一家交易公司工作：
Suppose Anthony and Susan work in same trading house:

- Anthony 可以讀取： 銀行 1 的 CD，天然氣公司的 CD
- Susan 可以讀取： 銀行 2 的 CD，天然氣公司的 CD

如果 Anthony 可以寫入天然氣公司的 CD，那麼 Susan 就可以讀取它。
If Anthony could write to Gas' CD, then Susan can read it.

因此，間接地，她可以讀取來自銀行 1 的 CD 的資訊，這明顯是利益衝突。
Hence, indirectly, she can read information from Bank 1's CD, a clear conflict of interest.

中國牆*-屬性通過禁止 Anthony 寫入天然氣公司的 CD 來防止這種情況，因為他能讀取來自不同 CD（銀行 1 和天然氣公司）的資訊。
*The Chinese Wall -*property prevents this by forbidding Anthony from writing to Gas' CD because he can read from different CDs (Bank 1 and Gas)*.

與 Bell-LaPadula 比較 / Comparison to Bell-LaPadula

特性 / Aspect	中國牆策略	Bell-LaPadula 模型
基礎	基於訪問歷史	基於安全標籤
狀態變化	權限動態變化	權限靜態（基於標籤）
時間要素	跟踪過去的訪問	只關心當前狀態

BLP 的局限性： 無法跟踪隨時間的變化。

Susan 生病了，Anna 需要接替她的工作。中國牆的歷史記錄讓 Anna 知道她是否可以訪問某些檔案。BLP 無法捕捉這一點。
Susan becomes ill, Anna needs to take over. CW history lets Anna know if she can. No way for Bell-LaPadula to capture this.

與 Clark-Wilson 比較 / Comparison to Clark-Wilson

如果主體和行程是可互換的，一個人可以使用多個行程來違反 CW 簡單安全條件，但仍然符合 Clark-Wilson 模型。
If subjects and processes are interchangeable, a single person could use multiple processes to violate CW-simple security condition, but would still comply with Clark-Wilson Model.

如果主體是特定的人，並包括該主體執行的所有行程，那麼與中國牆策略一致。
If subject is a specific person and includes all processes the subject executes, then consistent with Clark-Wilson Model.

小測試 / Test

1. 一個金融分析師已經訪問過"匯豐銀行"的檔案。根據中國牆策略，她現在還能訪問"渣打銀行"的檔案嗎？為什麼？
   A financial analyst has already accessed files from "HSBC". According to the Chinese Wall policy, can she now access files from "Standard Charter"? Why or why not?

2. 同一個分析師還能訪問"福特汽車"的檔案嗎？為什麼？
   Can the same analyst access files from "Ford Motor"? Why or why not?

3. 中國牆*-屬性主要防止什麼風險？
   *What risk does the Chinese Wall -property primarily prevent?

4. 中國牆策略與 BLP 和 Biba 模型最根本的區別是什麼？
   What is the most fundamental difference between the Chinese Wall policy and models like BLP and Biba?

答案與解析 / Answers and Explanations

1. **訪問同一 COI 類的不同公司**
   答案： 不能
   解析： 因為匯豐銀行和渣打銀行屬於同一個利益衝突類（都是銀行）。一旦分析師訪問了其中一個公司的資訊，她就被永久禁止訪問同一 COI 類中任何其他公司的資訊，以防止利益衝突。
   Answer: No
   Explanation: Because HSBC and Standard Charter belong to the same Conflict of Interest class (both are banks). Once the analyst accesses one company's information, she is permanently blocked from accessing any other company in the same COI class to prevent conflicts of interest.

2. **訪問不同 COI 類的公司**
   答案： 可以
   解析： 因為福特汽車屬於不同的利益衝突類（汽車製造，而不是銀行）。中國牆規則只限制對同一 COI 類內其他公司的訪問，不影響對其他行業公司的訪問。
   Answer: Yes
   Explanation: Because Ford Motor belongs to a different Conflict of Interest class (automotive manufacturing, not banking). The Chinese Wall rules only restrict access to other companies within the same COI class, not to companies in different industries.

3. **-屬性的目的**
   答案： 主要防止間接的資訊洩露和利益衝突。通過限制寫操作，確保使用者不能將自己從一個公司獲得的資訊通過共享區域（如他們都有權訪問的第三個公司的資料集）傳遞給可以訪問另一個競爭公司資訊的使用者。
   Answer: It primarily prevents indirect information leakage and conflicts of interest. By restricting write operations, it ensures a user cannot pass information obtained from one company to a user who has access to a competing company's information via a shared area (like a third company's dataset that both can access).

4. **最根本的區別**
   答案： 最根本的區別是中國牆策略是動態的和基於歷史的，而 BLP 和 Biba 是靜態的和基於狀態的。在中國牆中，權限隨著使用者的操作而改變；在 BLP/Biba 中，權限由預先分配的安全標籤固定決定。
   Answer: The most fundamental difference is that the Chinese Wall policy is dynamic and history-based, while BLP and Biba are static and state-based. In Chinese Wall, permissions change as the user acts; in BLP/Biba, permissions are fixed by pre-assigned security labels.

Role-based Access Control

中英雙語解釋 / Bilingual Explanation

基於角色的訪問控制介紹 / Introduction to Role-Based Access Control

RBAC 是一個廣泛使用的安全框架，尤其適用於商業環境。
RBAC is a widely used security framework claimed to be especially appropriate for commercial settings.

核心思想： RBAC 將權限與組織內的職能/工作/角色關聯起來，而不是直接分配給具體的主體。
RBAC associates permissions with functions/jobs/roles within an organization, unlike access

control policies that assign permissions directly to subjects.

- 角色是工作職能的集合。
  A role is a collection of job functions.
- 銀行中的角色可能包括： 行長、經理、培訓師、出納員、審計員、清潔工等。
  Roles within a bank might include: president, manager, trainer, teller, auditor, janitor, etc.

角色與事務 / Roles and Transactions

在 RBAC 模型中：
In the RBAC model:

個體擁有：

- **授權角色集：** 該個體被允許在不同時間擔任的角色。
  A set of authorized roles: which it is allowed to fill at various times.
- **活躍角色集：** 該個體當前正在擔任的角色。
  A set of active roles: which it currently occupies.

角色擁有相關的事務集： 這些是擔任該角色的人被允許執行的活動。
Roles have an associated set of transactions: which are the activities that someone in that role is permitted to carry out.

- **事務集可以是組織特定的：** 開戶、兌現支票、轉帳等。
  The set of transactions can be organization specific: open an account, cash a check, transfer funds, etc.

三條主要的 RBAC 規則 / Three Primary RBAC Rules

1. **角色分配：**
   主體只有在擁有活躍角色時才能執行事務。
   Role assignment: A subject can execute a transaction only if the subject has an active role.

2. **角色授權：**
   主體的活躍角色必須是該主體的授權角色之一。
   Role authorization: A subject's active role must be an authorized role for that subject.

3. **事務授權：**
   主體只有當事務被授權給該主體的某個活躍角色時才能執行該事務。
   Transaction authorization: A subject can execute a transaction only if the transaction is

authorized for one of the subject's active roles.

注意： 一個主體可以擁有多個角色。
Note that a subject can have multiple roles.

例子： 一個銀行行長也可能擔任出納員的角色。
Example: A bank president might also act as a teller.

角色包含與職責分離 / Subsumption and Separation of Duty

角色包含： 一個角色可能包含另一個角色。
*Subsumption: One role may subsume another.

- **含義：** 擁有角色 r_j 的人至少可以執行角色 r_i 的所有功能。
  Meaning that anyone having role r_j can do at least the functions of r_i.
- **例子：** 培訓師可以執行受訓者的所有動作，以及一些其他動作。
  Example: A trainer can perform all of the actions of a trainee, as well as some others.

職責分離： RBAC 也可以建模職責分離。
*Separation of Duty: RBAC can also model separation of duty.

- **要求：** 一個人不能同時擔任角色 r_1 和 r_2。
  One individual cannot assume both roles r_1 and r_2.
- **例子：** 如果出納員是某人的授權角色之一，那麼審計員就不能是。
  Example: If teller is among S's authorized roles, auditor cannot be.

RBAC 的優勢 / RBAC Advantages

- **更靈活：** RBAC 通常比標準的訪問控制策略更靈活。
  More flexible: RBAC is generally more flexible than standard access control policies.
- **易於管理：**
  - 角色"出納員"中的每個人都具有相同的權限。
    Everyone in role "teller" has the same permissions.
  - 添加新員工時，只需將其分配給適當的角色，而不是單獨分配數百個權限。
    When adding a new employee, simply assign them to the appropriate role rather than individually assigning hundreds of permissions.
- **符合業務邏輯：**
  - 權限與組織相關，例如"開戶"而不是"讀檔案"。
    Permissions are appropriate to the organization: "open an account" rather than "read a file".

- **承認多重職能：**
  - RBAC 認識到一個主體在組織內通常具有各種職能。
    RBAC recognizes that a subject often has various functions within the organization.
- **支援角色轉換：**
  - RBAC 允許主體在不同角色之間切換，而無需更改身份。
    RBAC allows a subject to transition between roles without having to change identities.

角色工程 / Role Engineering

定義： 定義角色並確定所需權限的過程。
*Role engineering: Defining roles and determining needed permissions.

- 當兩個使用 RBAC 的組織合併時經常用到。
  Often used when two organizations using RBAC merge.
- **挑戰：** 一個組織中的角色很少與另一個組織中的角色重疊，但工作職能經常重疊。
  Roles in one organization rarely overlap with roles in other, but job functions often do overlap.

角色挖掘： 分析現有角色、權限分配，以確定權限到角色的最優分配。
*Role mining: Analyzing existing roles, permission assignments to determine optimal assignment of permissions to roles.

- 這在理論上是 NP 完全問題，但在實踐中可以近似或產生最優解。
  NP-complete, but in practice optimal solutions can be approximated or produced.

小測試 / Test

1. 在 RBAC 中，"角色分配"、"角色授權"和"事務授權"這三條規則如何共同工作來控制訪問？
   How do the three RBAC rules - "Role Assignment", "Role Authorization", and "Transaction Authorization" - work together to control access?

2. 銀行出納員角色可能包含哪些典型的事務？為什麼使用"兌現支票"這樣的業務事務比使用"讀檔案 X"這樣的技術權限更好？
   What typical transactions might be included in a Bank Teller role? Why is using business transactions like "cash a check" better than using technical permissions like "read file X"?

3. 如何用 RBAC 實現"職責分離"，例如確保同一個人不能既建立採購訂單又授權付款？
   How can RBAC implement "Separation of Duty", for example, ensuring the same person cannot both create a purchase order and authorize its payment?

4. 當一家銀行收購另一家銀行時，在整合他們的 RBAC 系統時可能會遇到什麼挑戰？

What challenges might arise when integrating RBAC systems when one bank acquires another?

答案與解析 / Answers and Explanations

1. **三條規則的協同工作**
   答案： 這三條規則形成了一個分層控制。首先，主體必須有一個活躍角色（角色分配）。其次，該活躍角色必須是主體被授權擔任的（角色授權）。第三，主體嘗試執行的具體事務必須被包含在該活躍角色的授權事務集中（事務授權）。只有全部滿足，訪問才被允許。
   Answer: The three rules form a layered control. First, the subject must have an active role (Role Assignment). Second, that active role must be one the subject is authorized to assume (Role Authorization). Third, the specific transaction the subject attempts to execute must be within the set of transactions authorized for that active role (Transaction Authorization). Access is only granted if all three conditions are met.

2. **業務事務 vs 技術權限**
   答案： 出納員角色的事務可能包括："開戶"、"兌現支票"、"處理存款"、"查詢餘額"。
   使用業務事務更好的原因：

   - **更直觀：** 業務人員理解"兌現支票"的含義，但不一定理解"讀檔案 X"。
   - **更穩定：** 即使底層技術系統改變（如檔案位置變化），業務事務"兌現支票"仍然不變。
   - **更安全：** 減少了過度分配權限的風險（即分配了完成工作不需要的細粒度技術權限）。
     Answer: Transactions for a Teller role might include: "Open Account", "Cash Check", "Process Deposit", "Query Balance".
     Why business transactions are better:
   - More intuitive: Business people understand "cash a check" but not necessarily "read file X".
   - More stable: The business transaction "cash a check" remains the same even if the underlying technical system changes (e.g., file locations change).
   - More secure: Reduces the risk of over-privileging (assigning fine-grained technical permissions not needed for the job).

3. **實現職責分離**
   答案： 可以建立兩個角色："採購訂單建立者"和"付款授權者"。然後，在 RBAC 系統中配置一個職責分離約束，規定任何使用者都不能同時被分配這兩個角色。當試圖將第二個角色分配給已經擁有第一個角色的使用者時，系統會拒絕此操作。
   Answer: Create two roles: "Purchase-Order-Creator" and "Payment-Authorizer". Then,

configure a Separation of Duty constraint in the RBAC system stating that no user can be assigned both roles. The system will enforce this by refusing to assign the second role to a user who already has the first.

4. **整合 RBAC 系統的挑戰**
   答案： 主要挑戰是角色工程。兩家銀行可能對相似的工作職能使用不同的角色名稱和權限集合。例如，一家銀行的"客戶服務代表"角色可能與另一家銀行的"個人銀行家"角色職責重疊但權限不同。整合需要分析兩家銀行的實際工作流程，定義一套新的、統一的標準角色，並將員工映射到這些新角色上。這是一個複雜的過程，稱為"角色挖掘"。
   Answer: The main challenge is role engineering. The two banks likely use different role names and permission sets for similar job functions. For example, one bank's "Customer Service Representative" role might overlap with the other bank's "Personal Banker" role but have different permissions. Integration requires analyzing the actual workflows of both banks, defining a new, unified set of standard roles, and mapping employees to these new roles. This complex process is known as "role mining".

Storing the Access Control Matrix

中英雙語解釋 / Bilingual Explanation

訪問控制矩陣的儲存問題 / The Problem of Storing the Access Control Matrix

回憶： 任何訪問控制策略都可以用訪問控制矩陣來表示。
Recall that any access control policy can be represented by an access control matrix.

ACM 明確表示了每個主體對每個客體被允許的每一次訪問。
The ACM gives an explicit representation of every access permitted by every subject to every object.

問題： 對於大型系統，顯式儲存 ACM 會極其昂貴且低效。
Problem: For large systems, storing an explicit ACM is extremely expensive and inefficient.

三種常見的實現方式 / Three Common Implementation Alternatives

有三種常見的替代方案來儲存和計算訪問權限：
Three common alternatives exist for storing and computing access permissions:

1. 基於規則的訪問控制
   Rule-Based Access Control
2. 訪問控制列表

Access Control Lists (ACLs)

3. 基於能力的系統
   Capability-Based Systems

1. 基於規則的訪問控制 / Rule-Based Access Control

原理： 維護一組規則，根據主體和客體的屬性動態計算訪問權限。
Principle: Maintain a set of rules to compute access permissions "on the fly" based on attributes of subjects and objects.

例子： BLP 和 Biba 模型就是典型的例子。
Examples: BLP and Biba models are perfect examples.

系統不儲存龐大的矩陣，而是儲存簡單的規則：
The system doesn't store a huge matrix but stores simple rules:

- BLP 規則： "允許讀，僅當主體等級 ≥ 客體等級"
  BLP Rule: "Allow read only if subject level ≥ object level"
- Biba 規則： "允許寫，僅當主體完整性等級 ≤ 客體完整性等級"
  Biba Rule: "Allow write only if subject integrity level ≤ object integrity level"

優點： 緊湊，易於理解策略意圖。
缺點： 每次訪問都需要計算，可能影響效能。

2. 訪問控制列表 / Access Control Lists (ACLs)

原理： 將權限與客體儲存在一起。
Principle: Store the permissions with the objects of the system.

結構： 包含形式為 的對的列表，列出了主體 S 當前對該客體持有的權限集 P。
Structure: It contains pairs of the form , listing the set of permissions P that subject S currently holds to the object.

現實世界的例子：

- **Unix/Linux 檔案系統：** 每個檔案都有 ACL，包含所有者、組和其他使用者的讀/寫/執行權限。
  Unix/Linux file system: Each file has an ACL with read/write/execute permissions for owner, group, and others.
- **Windows NTFS：** 更複雜的 ACL，可以指定單個使用者或組的權限。
  Windows NTFS: More complex ACLs that can specify permissions for individual users or

groups.

工作原理： 當主體請求訪問客體時，系統檢查該客體的 ACL，查看該主體是否被授予了請求的權限。

3. 基於能力的系統 / Capability-Based Systems

原理： 將權限與主體儲存在一起。
Principle: Store the permissions with subjects rather than objects.

結構： 每個主體維護一個形式為 的對的集合，表示主體 S 當前對客體 O 執行訪問 A 的權限。
Structure: Each subject maintains a collection of pairs meaning the subject has permission to perform access A to object O.

關鍵特性： 能力本身就是權限的證明。
Key Feature: A capability is de facto evidence of permission.

要獲得訪問權，主體必須出示相應的能力。
To obtain access, the subject must present an appropriate capability.

因此，能力也是一種票據。
Thus, a capability is also a type of ticket.

保護能力 / Protecting Capabilities

由於擁有能力就意味著擁有權限，因此防止能力被偽造或篡改至關重要。
Since possession of a capability is proof of permission, it's necessary to ensure capabilities can't be forged or altered.

歷史上使用的保護方法：
Historical approaches to protect capability integrity:

- **特殊標記：** 擴展每個記憶體位置，用一個附加位指示該位置是否包含能力。
  Extend each memory location with an additional bit indicated whether or not the location contains a capability.
- **操作系統獨佔：** 只有操作系統可以操作能力。
  Only the OS can manipulate capabilities.
- **受保護記憶體：** 將能力儲存在特殊保護的記憶體中。
  Store capabilities in specially protected memory.

能力傳遞： 許多基於能力的系統也允許在受控情況下將能力從一個主體傳遞給另一個主體。

Many capability-based systems also permit passing capabilities from one subject to another, under controlled circumstances.

ACL vs 能力：比喻 / ACLs vs Capabilities: An Analogy

方面 / Aspect	訪問控制列表	基於能力
比喻 / Analogy	俱樂部會員名單	音樂會門票
工作原理	保安在門口有一份被允許進入的人員名單。當你到達時，保安檢查名單上是否有你的名字。	你持有一張門票。當你到達時，你出示門票作為進入的證明。
權限儲存	權限儲存在客體（俱樂部）處。	權限由主體（你）持有。
訪問檢查	需要中央驗證（保安檢查名單）。	無需中央驗證（檢票員只需確認門票真實）。

小測試 / Test

1. 在基於規則的系統中（如 BLP），當主體請求讀取客體時，系統如何決定是否允許訪問？
   In a rule-based system like BLP, how does the system decide whether to allow access when a subject requests to read an object?

2. 在訪問控制列表系統中，權限資訊儲存在哪裡？當使用者嘗試打開檔案時，系統具體檢查什麼？
   In an ACL-based system, where is the permission information stored? What exactly does the system check when a user tries to open a file?

3. 基於能力的系統的一個關鍵安全風險是什麼？系統如何減輕這種風險？
   What is a key security risk in capability-based systems? How do systems mitigate this risk?

4. 哪種實現方式最適合實現 BLP 模型？哪種最適合實現 RBAC 模型？為什麼？
   Which implementation approach is best suited for implementing the BLP model? Which is best suited for implementing RBAC? Why?

答案與解析 / Answers and Explanations

1. **基於規則的訪問決策**

答案： 系統動態計算權限。它查看主體的安全標籤和客體的安全標籤，然後應用預定義的規則。例如，對於 BLP，它檢查"主體的等級是否 ≥ 客體的等級？"。如果規則滿足，則允許訪問；否則拒絕。
Answer: The system dynamically computes permissions. It looks at the subject's security label and the object's security label, then applies predefined rules. For BLP, it checks "Is the subject's level ≥ the object's level?". If the rule is satisfied, access is granted; otherwise, it's denied.

2. **ACL 系統的訪問檢查**
答案： 權限資訊儲存在每個客體的訪問控制列表中。當使用者嘗試打開檔案時，系統檢索該檔案的 ACL，並在列表中查找該使用者的條目（或使用者所屬組的條目），檢查請求的權限（如讀、寫）是否被授予。
Answer: Permission information is stored in the Access Control List associated with each object. When a user tries to open a file, the system retrieves the file's ACL and looks for an entry for that user (or groups the user belongs to) to see if the requested permission (e.g., read, write) is granted.

3. **能力系統的風險與緩解**
答案： 關鍵風險是能力的偽造或篡改。因為擁有能力就意味著擁有訪問權，如果攻擊者能夠偽造能力，他們就可以獲得未經授權的訪問。
緩解措施： 使用硬體支援（特殊標記位）、將能力儲存在操作系統控制的受保護記憶體中、使用密碼學技術對能力進行簽名以防止篡改。
Answer: A key risk is the forgery or alteration of capabilities. Since possession of a capability grants access, if an attacker can forge capabilities, they can gain unauthorized access.
Mitigation: Use hardware support (special tag bits), store capabilities in OS-controlled protected memory, or use cryptographic techniques to sign capabilities to prevent tampering.

4. **模型與實現方式的匹配**

   ◦ **BLP 模型 → 基於規則：** BLP 的策略由簡單的、全域的規則定義（"無向上讀"）。基於規則的實現非常自然，因為規則可以直接編碼這些約束。為每個檔案-使用者對儲存 ACL 或將能力分配給每個使用者會非常低效。
   ◦ **RBAC 模型 → ACL 或能力：** RBAC 將權限與角色關聯。這可以很自然地實現為：ACL 按角色命名（而不是按使用者名稱），或者使用者持有代表其角色的能力。當使用者啟動角色時，他們獲得該角色的能力。
   ◦ BLP Model → Rule-Based: BLP's policy is defined by simple, global rules ("no read up"). A rule-based implementation is natural because the rules can directly encode

these constraints. Storing ACLs for every file-user pair or capabilities for every user would be very inefficient.

- RBAC Model → ACLs or Capabilities: RBAC associates permissions with roles. This can be naturally implemented as: ACLs that name roles (instead of usernames), or users holding capabilities that represent their roles. When a user activates a role, they acquire the capabilities of that role.

Clinical Information Systems Security Policy

中英雙語解釋 / Bilingual Explanation

臨床資訊系統安全策略概述 / Clinical Information Systems Security Policy Overview

目標： 用於保護醫療記錄。
Intended for medical records.

核心關注點： 在這裡，利益衝突不是關鍵問題，關鍵是患者保密性、記錄和註釋者的認證以及完整性。
Core Concerns: Patient confidentiality, authentication of records and annotators, and integrity are critical here (conflict of interest is not).

實體定義：
Entities:

- **患者：** 醫療記錄的主體（或其代理人）。
Patient: subject of medical records (or agent).
- **個人健康資訊：** 關於患者健康或治療的資料，能夠識別患者身份。
Personal health information: data about patient's health or treatment enabling identification of patient.
- **臨床醫生：** 在執行工作時訪問個人健康資訊的醫療保健專業人員。
Clinician: health-care professional with access to personal health information while doing job.

核心原則 / Core Principles

該策略基於醫學倫理和實踐需求制定了一系列原則：
The policy establishes principles derived from medical ethics and practical needs:

1. **訪問原則 1 - 訪問控制列表**

   Access Principle 1 - Access Control List

    - 每個醫療記錄都有一個訪問控制列表，列出了可以讀取和追加資訊到該記錄的個人或組。
      Each medical record has an access control list naming who may read and append information.
    - 系統必須將訪問限制在 ACL 中標識的人員。
      The system must restrict access to those on the ACL.

2. **訪問原則 2 - 責任臨床醫生**
   Access Principle 2 - Responsible Clinician

    - ACL 上的一位臨床醫生必須有權將其他臨床醫生添加到 ACL 中。
      One clinician on the ACL must have the right to add others to the ACL.
    - 這被稱為責任臨床醫生，對患者的護理和治療負總體責任。
      Called the responsible clinician with overall responsibility for care.

3. **訪問原則 3 - 患者知情同意**
   Access Principle 3 - Patient Notification and Consent

    - 每當患者的醫療記錄被打開時，責任臨床醫生必須將 ACL 上的姓名通知患者。
      The responsible clinician must notify the patient of names on the ACL when record is opened.
    - 除法定情況或緊急情況外，必須獲得患者的同意。
      Must obtain the patient's consent except in emergencies/statutory situations.

4. **訪問原則 4 - 審計日誌**
   Access Principle 4 - Audit Logging

    - 必須記錄訪問醫療記錄的臨床醫生姓名、日期和時間。
      Must record name, date, and time of access.
    - 刪除操作也必須保留類似資訊。
      Similar information for deletions.

5. **建立原則 - 記錄建立**
   Creation Principle - Record Creation

    - 臨床醫生可以建立記錄，將自己和患者放在 ACL 上。
      A clinician may open a record, with themselves and the patient on the ACL.
    - 如果是轉診結果，轉診臨床醫生也可能在 ACL 上。

If from a referral, the referring clinician may also be on the ACL.

6. **刪除原則 - 受控刪除**
   Deletion Principle - Controlled Deletion

   - 在適當時間過去之前，不能從醫療記錄中刪除臨床資訊。
     Clinical information cannot be deleted until appropriate time has passed.

關鍵安全原則 / Key Security Principles

7. **限制原則 - 資訊流控制**
   Confinement Principle - Information Flow Control

   - 一個醫療記錄中的資訊可以追加到另一個醫療記錄中，當且僅當第二個記錄的 ACL 是第一個記錄 ACL 的子集。
     Information may be appended to another record only if the second record's ACL is a subset of the first's.
   - **目的：** 防止資訊洩露給未授權使用者。
     Purpose: Prevents information leakage to unauthorized users.

8. **聚合原則 - 防止資料聚合**
   Aggregation Principle - Preventing Data Aggregation

   - 必須採取有效措施防止患者資料的聚合。
     Measures for preventing aggregation of patient data must be effective.
   - 如果有人要被添加到患者的記錄 ACL 中，必須通知患者，特別是當該人可以訪問大量醫療記錄時。
     Patient must be notified if anyone with access to many records is added to ACL.
   - **擔憂：** 腐敗的調查員可能訪問大量記錄，關聯它們，並發現個人的私人資訊用於惡意目的（如敲詐）。
     Fear: A corrupt investigator may correlate many records to discover private information for blackmail.

執行要求 / Enforcement Requirements

任何處理醫療記錄的計算系統必須有一個強制執行前述原則的子系統。
Any system must have a subsystem that enforces the principles.

這種執行的有效性必須接受獨立審計員的評估。
The effectiveness must be subject to evaluation by independent auditors.

與其他模型比較 / Comparison to Other Models

與 Bell-LaPadula 比較：
Compare to Bell-LaPadula:

- 限制原則在模型中的實體上施加了格結構，類似於 BLP。
  Confinement Principle imposes lattice structure similar to BLP.
- **關鍵區別：** CISS 關注被訪問的物件，而 BLP 關注訪問物件的主體。
  Key difference: CISS focuses on objects being accessed, while BLP on subjects accessing objects.

與 Clark-Wilson 比較：
Compare to Clark-Wilson:

- CDIs 是醫療記錄。
  The CDIs are medical records.
- TPs 是更新記錄、訪問控制列表的功能。
  The TPs are functions updating records, access control lists.
- IVPs 認證： The IVPs certify:
  - 被識別為臨床醫生的人確實是臨床醫生
    A person identified as a clinician is a clinician;
  - 臨床醫生驗證或已經驗證了醫療記錄中的資訊
    A clinician validates information in the record;
  - 當需要通知某人時，通知確實發生
    When someone is to be notified, notification occurs;
  - 當需要同意時，在獲得同意前操作不能進行
    When consent is needed, operation waits until consent is obtained.

小測試 / Test

1. 限制原則如何防止資訊洩露？請用具體例子說明。
   How does the Confinement Principle prevent information leakage? Provide a concrete example.

2. 為什麼聚合原則在醫療記錄系統中特別重要？它防止什麼具體威脅？
   Why is the Aggregation Principle particularly important in medical record systems? What specific threat does it prevent?

3. CISS 策略中的"責任臨床醫生"角色與 Clark-Wilson 模型中的什麼概念相似？
   How does the "responsible clinician" role in CISS relate to concepts in the Clark-Wilson

model?

4. 與 BLP 模型相比，CISS 策略在關注點上有什麼根本不同？
   What is the fundamental difference in focus between the CISS policy and the BLP model?

答案與解析 / Answers and Explanations

1. **限制原則的防洩露機制**
   答案： 限制原則通過控制資訊流動的方向來防止洩露。它要求目標記錄的 ACL 必須是源記錄 ACL 的子集。
   例子： 記錄 A 的 ACL 包含 {Dr. Smith, Dr. Jones, Patient X}。記錄 B 的 ACL 包含 {Dr. Smith, Patient X}（這是記錄 A ACL 的子集）。因此，允許將資訊從 A 追加到 B。但記錄 C 的 ACL 包含 {Dr. Brown, Patient X}（這不是 A 的子集），則不允許從 A 向 C 追加資訊，因為 Dr. Brown 未被授權訪問記錄 A。
   Answer: The Confinement Principle prevents leakage by controlling the direction of information flow. It requires that the target record's ACL must be a subset of the source record's ACL.
   Example: Record A's ACL is {Dr. Smith, Dr. Jones, Patient X}. Record B's ACL is {Dr. Smith, Patient X} (a subset of A's ACL). So, appending info from A to B is allowed. But Record C's ACL is {Dr. Brown, Patient X} (not a subset of A's), so appending from A to C is forbidden because Dr. Brown isn't authorized for Record A.

2. **聚合原則的重要性**
   答案： 聚合原則防止通過關聯多個記錄來推斷敏感資訊的威脅。單個醫療記錄可能看似無害，但當攻擊者能訪問成千上萬個記錄時，他們可以通過資料探勘發現模式（如某種基因疾病在特定人群中的流行率），這些聚合資訊可能被用於歧視、敲詐或保險詐騙。通知患者有人（特別是能訪問大量記錄的人）被添加到其 ACL 中，增加了透明度，並允許患者質疑可疑的訪問。
   Answer: It prevents the threat of inferring sensitive information by correlating multiple records. A single medical record might seem harmless, but when an attacker accesses thousands, they can data-mine for patterns (e.g., prevalence of a genetic disease in a population). This aggregated information could be used for discrimination, blackmail, or insurance fraud. Notifying the patient when someone with broad access is added to their ACL increases transparency and allows the patient to challenge suspicious access.

3. **責任臨床醫生與 Clark-Wilson 的關聯**
   答案： "責任臨床醫生"的角色類似於 Clark-Wilson 模型中認證和維護"認證關係"與"允許關係"的信任實體。具體來說，責任臨床醫生有權修改 ACL（相當於更改"允許關係"），這類似於 Clark-Wilson 規則 E4 中規定的"只有 TP 的認證者才能更改與該 TP 關聯的實體列

表"。
Answer: The "responsible clinician" role is analogous to the trusted entity in Clark-Wilson that certifies and maintains the "certified relations" and "allowed relations". Specifically, the responsible clinician has the authority to modify the ACL (changing the "allowed relation"), similar to Clark-Wilson Rule E4: "Only the certifier of a TP may change the list of entities associated with that TP."

4. **CISS 與 BLP 的根本區別**
   答案： 根本區別在於關注的焦點。BLP 是以主體為中心的——它關注控制哪個主體可以訪問什麼資訊，主要防止資訊從高密級主體流向低密級主體。CISS 是以物件為中心的——它關注保護單個醫療記錄的機密性和完整性，控制誰能訪問特定的記錄，以及資訊如何在記錄之間流動。CISS 更關心患者隱私和資料的正確性，而不是主體的安全等級。
   Answer: The fundamental difference is the focus. BLP is subject-centric – it focuses on controlling which subject can access what information, primarily preventing flow from high to low clearance. CISS is object-centric – it focuses on protecting the confidentiality and integrity of individual medical records, controlling who can access a specific record and how information flows between records. CISS is more concerned with patient privacy and data correctness than subject security levels.