سوال ۱)

الف)

از جمله مشکلات کاربردی می توان گفت از آنجایی که جریان اطلاعات در BLP از پایین به بالا است و هیچ گونه اطلاعاتی نمی تواند ازسطوح بالا به سطوح پایین انتقال یابد لذا عملا امکان دستور و فرمان از سطوح بالا به پایین ممکن نیست و این امر کاربرد را دچار مشکل می سازد. مثلا اگر یک colonel در سطح (top secret, {nuc, eur}) باشد و یک major در سطح (secret, {eur}) و دستور دادن را معادل نوشتن و دریافت دستور را معادل خواندن بدانیم آن گاه colonel نمی تواند هیچ فرمانی به major بدهد که مطلوب نیست.

از جمله مشکلات امنیتی می توان گفت یک روش نشت اطلاعات و انتقال آن ها از سطوح بالا به پایین وجود دارد و این روش اصطلاحا کانال نهان نامیده می شود و به این صورت است : از آنجایی که subject های سطوح پایین می توانند در سطوح بالا تر یا در سطح خود از خود بنویسند حال اگر یک subject قصد ایجاد یک دایرکتوری یا فایل با نامی معین را جهت نوشتن داشته باشد چناچه یک دایرکتوری یا فایل به آن نام قبلا در آن سطح موجود باشد ارور دریافت می کند در غیر اینصورت فایل با موفقیت ایجاد می شود.

از این ویژگی می توان استفاده کرد و دو subject در دو سطح متفاوت در بازه های زمانی معین نسبت به ایجاد فایلی با نام یکسان در یکی از سطوح مشخص بالا اقدام می کنند حال مثلا اگر subject سطح بالا قصد نمایش بیت ۰ را داشته باشد فایل با نام معین را در سطح مذکور ایجاد می کند و یا اگر قصد نمایش بیت ۱ را داشته باشد فایل با نام معین را در سطح مذکور حذف می کند به این ترتیب subject سطح پایین پس از آن اقدام به ایجاد فایل می کند و اگر به ارور بر خورد کرد به معنای ۰ و در غیر اینصورت به معنای ۱ تلقی می شود و این ترتیب می توانند در حالی که در سطوح امنیتی متفاوت قرار دارند باهم تبادل اطلاعات کنند ک موجب نشت اطلاعات از سطح بالا به سطوح پایین تر می شود.

ب)

خیر در این روش تنها مالک اولیه فایل تعیین می کند چه کسانی به فایل دسترسی داشته باشند.

ج)

خیر به طورکلی مدل biba برمبنای integrity عمل می کند و نه confidentiality.

سوال ۲)

الف)

می توان از روش ماتریس کنترل دسترسی و یا capability list استفاده کرد.

ب)

در این حالت کرنل وارد عمل شده و کنترل می کند که آیا آرگومان هایی که در این فراخوانی استفاده شده آرگومان های درستی هست یا خیر. و آیا شرایط فراخوانی امن هست یا خیر اگر امن باشد کرنل اجازه ادامه فراخوانی را می دهد در غیر اینصورت از ادامه کار جلوگیری می کند.

ج)

Fail-safe-default : این اصل بیان می کند اگر قرار است در مجوز دهی پیش فرضی در نظر گرفته شود آن پیش فرض باید امن باشد مثلا اگه قرار است به طور پیش فرض مجوز ها را برای یک object جدید ایجاد کنیم به صورت پیش فرض به آن object هیچ مجوزی داده نشود چون این یک پیش فرض امن است.

سوال ۳)

```
command Grant.read.file(Si, O, Sj)
        if own in A[Si, O] and r in A[Si, O]
        then
                enter r into A[Sj, O]
end

command Grant.write.file(Si, O, Sj)
        if own in A[Si, O] and w in A[Si, O]
        then
                enter w into A[Sj, O]
end

command Revoc.execution.file(Si, O, Sj)
        if own in A[Si, O]
        then
                delete x from A[Sj, O]
end

command Create.file (Si, O)
        create object O
        enter own into A[Si, O]
        enter r into A[Si, O]
        enter w into A[Si, O]
        enter x into A[Si, O]
end
```

الف)

Create.file (Subj1, Obj4): این دستور باعث ایجاد یک ستون جدید در ماتریس برای Object Obj4 و اضافه شدن دسترسی های Owner, Read, Write, Execution برای Subj1 نسبت به Obj4 میشود. بنابراین ماتریس تغییر میکند.

Grant.write.file(Subj1, Obj3, Subj2): با توجه به اینکه Subj1 مالک Obj3 نیست این دستور اجرا نمیشود. بنابراین ماتریس تغییری نمیکند.

Grant.read.file(Subj1, Obj4, Subj2): چون Subj1 مالک Obj4 است و دسترسی خواندن بر روی فایل Obj4 هم دارد دستور اجرا می‌شود و اجازه ی خواندن Obj4 به Subj1 داده میشود. بنابراین ماتریس تغییر میکند.

Revoc.execution.file(Subj1, Obj3, Subj3): با توجه به اینکه Subj1 مالک Obj3 نیست این دستور اجرا نمیشود. بنابراین ماتریس تغییری نمیکند.

ب)

	Obj1	Obj2	Obj3	Obj4
Subj1	OWRX	R	RWX	OWRX
Subj2	R	RW	R	R
Subj3	RW	ORW	OWRX	

ج)

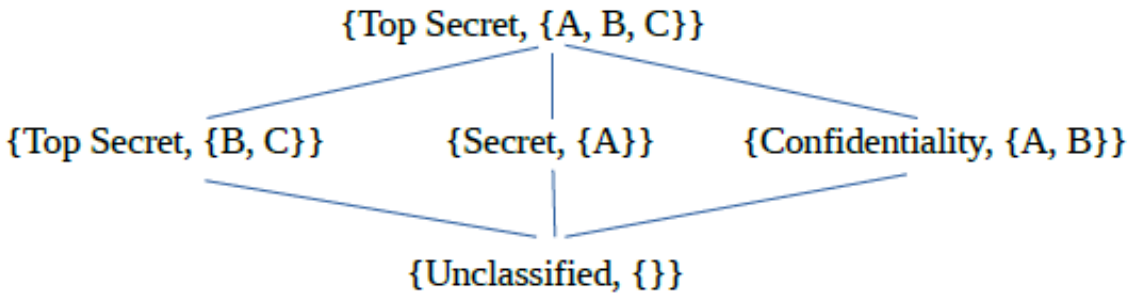
ACL
Obj1 = {(Subj1, OWRX) (Subj2, R) (Subj3, RW)}
Obj2 = {(Subj1, R) (Subj2, RW) (Subj3, ORW)}
Obj3 = {(Subj1, RWX) (Subj2, R) (Subj3, OWRX)}
Obj4 = {(Subj1, OWRX) (Subj2, R)}

C-List
Subj1 = {(Obj1, OWRX) (Obj2, R) (Obj3, RWX) (Obj4, OWRX)}
Subj2 = {(Obj1, R) (Obj2, RW) (Obj3, R) (Obj4, R)}
Subj3 = {(Obj1, RW) (Obj2, ORW) (Obj3, OWRX)}

سوال ۴)

subject s در صورتی میتواند object o را بخواند که L(s) dom L(o) و s مجوز خواندن o را داشته باشد.
subject s در صورتی میتواند در object o بنویسد که L(o) dom L(s) و s مجوز نوشتن o را داشته باشد.
الف) خیر. چون مجموعه ی {A, B} زیر مجموعه ی مجموعه ی {B, C} نیست اجازه ی خواندن وجود ندارد. همچنین چون Confidential < Top Secret و مجموعه ی {B, C} زیر مجموعه ی مجموعه ی {A, B} نیست اجازه ی نوشتن نیز وجود ندارد.
ب) خیر. چون Secret < Top Secret و مجموعه ی {B, C} زیر مجموعه ی مجموعه ی {A} نیست اجازه ی خواندن وجود ندارد. همچنین چون مجموعه ی {A} زیر مجموعه ی مجموعه ی {B, C} نیست اجازه ی نوشتن نیز وجود ندارد.
ج) خیر. چون Confidential < Secret اجازه ی خواندن وجود ندارد. همچنین چون مجموعه ی {A, B} زیر مجموعه ی مجموعه ی {A} نیست اجازه ی نوشتن نیز وجود ندارد.
د) چون سناریو یک مدل BLP است جریان اطلاعات از پایین به بالا است. (هر subj اجازه ی خواندن obj های پایین‌تر از خود و نوشتن obj های بالاتر از خود را دارد.)

ه)



subject s در صورتی میتواند object o را بخواند که i(o) dom i(s) و s مجوز خواندن o را داشته باشد.
subject s در صورتی میتواند در object o بنویسد که i(s) dom i(o) و s مجوز نوشتن o را داشته باشد.
الف) خیر. چون مجموعه ی {B, C} زیر مجموعه ی مجموعه ی {A, B} نیست اجازه ی خواندن وجود ندارد. همچنین چون Trusted < Very Trusted و مجموعه ی {A, B} زیر مجموعه ی مجموعه ی {B, C} نیست اجازه ی نوشتن نیز وجود ندارد.
ب) خیر. چون مجموعه ی {A} زیر مجموعه ی مجموعه ی {B, C} نیست اجازه ی خواندن وجود ندارد. همچنین چون Slightly Trusted < Trusted و مجموعه ی {B, C} زیر مجموعه ی مجموعه ی {A} نیست اجازه ی نوشتن نیز وجود ندارد.
ج) چون Trusted < Very Trusted و مجموعه ی {A} زیر مجموعه ی مجموعه ی {A, B} است، i(s) dom i(o). بنابراین دسترسی نوشتن وجود دارد. ولی دسترسی خواندن وجود ندارد.
د) چون سناریو یک مدل Biba است جریان اطلاعات از بالا به پایین است. (هر subj اجازه ی خواندن obj های بالاتر از خود و نوشتن obj های پایین تر از خود را دارد.)

سوال ۵)

الف)

معایب: در روش شمیر حقوق مختلف از هم جدا نمی شوند زیرا یک subject با داشتن کلید یک object به همه حقوق آن به طور همزمان دسترسی پیدا می کند مثلا حق خواندن ، نوشتن و ...

می توانیم برای حل این مشکل کلید های جداگانه ای برای نوشتن یک object ، خواندن آن و ... در نظر بگیریم.

مزایا: عمل revocation به سادگی قابل پیاده سازی است کافی است که اگر قصد سلب حق دسترسی k امین subject به یک object همچون O را داریم Ek (O) را از بین ببریم.

ب) مقدار threshold برابر ۳ (با توجه به اینکه درجه ی معادله برابر ۲ است.) و مقدار secret برابر ۱۵۲۳ است.

$y = 5024x^2 + 1234x + 1523$

$t_1 = (0, 1523)$

$t_2 = (-1, 5313)$

$t_3 = (1, 7781)$

$l_0 = \frac{x-(-1)}{0-(-1)} \times \frac{x-1}{0-1} = -x^2 + 1$

$l_1 = \frac{x-0}{(-1)-0} \times \frac{x-1}{(-1)-1} = \frac{1}{2}(x^2 - x)$

$l_2 = \frac{x-0}{1-0} \times \frac{x-(-1)}{1-(-1)} = \frac{1}{2}(x^2 + x)$

$y = y_0l_0 + y_1l_1 + y_2l_2 = 5024x^2 + 1234x + 1523$