平成１８年度　春期　テクニカルエンジニア（情報セキュリティ）　午後Ⅰ問題

問１　Web サイトのセキュリティに関する次の記述を読んで，設問 1～4 に答えよ。

A 社は，主として事業所向けに商品を販売している，従業員数 80 名の文房具卸売会社である。A 社ではこれまで，インターネットを電子メールと Web の閲覧にしか使ってこなかったが，このたびインターネットを利用した受注 Web システム（以下，X システムという）を開発することに決めた。A 社には情報システムを開発する部署がないので，現在インターネット接続を管理している F 主任をリーダとして，コンピュータに詳しい数名の従業員で構成した開発チームが，開発を行うことになった。

〔X システムの基本設計〕

X システムでは，利用者はインターネットから Web サイトにアクセスし，商品を選択して発注すると，その利用者 ID，発注商品名及び数量が営業担当の従業員に電子メールで通知される。営業担当の従業員はその電子メールに従って，利用者あてに商品，納品書及び請求書を発送する。当初の利用者数は 1 日当たり 1,000 人と想定している。

F 主任は，開発に先立って，注意すべき事項について開発チーム内で話し合うことにした。次は，そのときの F 主任と開発チームの一員である G 君の会話である。

F 主任：X システムでは，画面遷移にセッション管理が必要だな。

G 君　：そうですね。セッション管理には，セッション ID を使う方式と，利用者 ID を使う方式があります。いずれの方式でも利用者のブラウザからサーバに利用者を識別するデータを送信しますが，この送信方式にはクッキーを使う方式，［　a　］フィールドを使う方式，URL のクエリストリングを使う方式があるようです。それぞれの特徴を整理すると，表 1 のようになります。利用者からのデータの送信には POST メソッドと GET メソッドの両方を使う予定ですが，送信方式はどの方式を用いるべきでしょうか。

表１　セッション管理用データの送信方式の特徴

方　式	各方式の特徴	
	利用できる環境	HTTP リクエストヘッダの Referer フィールドからの情報漏えいの可能性
（ア）クッキーを使う方式	ブラウザがクッキーをサポートしている必要がある。	［　b　］
（イ）［　a　］フィールドを使う方式	HTML フォームに限る。	なし （GET メソッド使用の場合は“あり”）
（ウ）URL のクエリストリングを使う方式	上記（ア），（イ）の方式のような制限はない。	［　c　］

F 主任：表 1 の各方式の特徴の中で，利用者からのデータ送信に用いる方法を考慮して，情報漏えいの可能性が最も低い方式を選ぶのがよいだろう。X システムの利用者の多くは事業所内からアクセスすると想定できるので，携帯電話からのアクセスは考慮しないことにしよう。

G 君　：分かりました。

F 主任：それから，クロスサイトスクリプティング脆弱性を悪用した攻撃など，利用者から入力されるデータは，悪意のある可能性がある。利用者からの入力データを含む Web ページを生成してブラウザに表示する場合には，①攻撃に利用される文字を表 2 のように置き換える処理が必要だ。セキュリティ機能にかかわる部分は一貫した考え方が必要だと思うので，まとめて G 君に考えて作成してもらいたい。

表２　文字の置換

元の文字	置換後の文字列
&	&amp;
D	e
>	>
'	&#39;
"	&quot;

G 君　：分かりました。これまで，本格的にセキュリティ機能の設計やコーディングを行ったことはないのですが，まず利用者認証とセッション管理について設計し，Perl でコーディングしてみようと思います。セッション管理については，コーディングの負担を減らし，短期間で開発するために利用者 ID を使う方式にします。

F 主任：その場合でも，セッション管理用データが不正に作り出せないような設計をするように。また，総当たり法による攻撃も想定する必要がある。

〔設計レビュー〕

3 日後，G 君は利用者 ID を使用したセッション管理用データ（以下，識別コードという）の仕様とサブルーチンの基本仕様を作成し，一部分をコーディングして開発チーム内でレビューを受けた。G 君が作成した，識別コードを使用したセッション管理の仕様を表 3 に，サブルーチンの基本仕様の一部を表 4 に，またコードの一部を図に示す。

表３　識別コードを使用したセッション管理の仕様（抜粋）

項　目	要　件
利用者 ID, パスワード	利用者 ID，パスワードの文字列長はともに 6 文字以上 12 文字以下とする。
識別コード	60 文字以上の英数字とする。
タイムアウト	前画面からの遷移までに 10 分以上経過している場合は，タイムアウトエラーとする。

表４　サブルーチンの基本仕様（抜粋）

ルーチン名	処理概要	引数	戻り値
clean_msg	引数の文字列中の攻撃に利用される文字を表 2 に従って置き換える。	1. 文字列	処理された文字列
make_sid	識別コードを作成する。	1. 利用者 ID（文字列）	識別コードを表す文字列
Encrypt	第 1 引数の文字列を第 2 引数の共通鍵で暗号化する。	1. 文字列（16 文字以内） 2. 128 ビットの共通鍵を表す文字列（32 文字）	暗号化された 32 文字の文字列
Now	現在時刻を秒単位で得る。	なし	YYYYMMDDhhmmss 形式によって現在時刻を表す 14 文字の文字列
make_hash	引数の文字列をハッシュ関数で計算した値を返す。	1. 文字列	ハッシュ値を表す 32 文字の文字列

```
sub make_sid {
     my ($user_ID) = @_;
     my $enc_key = "4f583bd08d45092ca12408c8437ef4";
     my $sid, $salt, $sid_time;

     $sid_time = now;
     $sid_time =~ /..$/;
     $salt = $&;
     $sid = encrypt($user_ID, $salt . $enc_key);
     $sid = $sid . $sid_time;
     $sid = $sid . make_hash($sid);
     return $salt . $sid;
}
```

図　G君が作成したコード（抜粋）

次は，レビューにおける F 主任と G 君の会話である。

F 主任：識別コードの仕様について説明してくれないか。

G 君　：はい。識別コードは，その生成時刻と利用者 ID からなる文字列を基本情報として構成され，利用者がログイン後に Web サーバにアクセスする都度，生成されます。生成時刻は，識別コードが Web サーバに返されたときにタイムアウトを判定するために付けています。利用者 ID は，識別コードが不正に作り出されることを防止するために，鍵長が 128 ビットの共通鍵暗号を使って暗号化しています。共通鍵は 128 ビットの固定値ではなく，そのうち 8 ビットを可変部分としました。8 ビットを可変とすることで，毎回異なる識別コードが生成されるので安全です。また，改ざんを検出するために，暗号化した利用者 ID 及びサーバの時刻情報に，それらのハッシュ値を連結しています。このため識別コードは，[f] 文字の利用者 ID 部分，[g] 文字の時刻部分，[h] 文字のハッシュ部分を連結した文字列に，共通鍵の可変部分を先頭に加えて，合計 [i] 文字で構成されます。

F 主任　：この識別コードの作り方では，暗号化された利用者 ID の部分には同じ文字列が現れることがある。また，②生成される識別コードは毎回異なるが，共通鍵の作り方から判断して安全性が高くなっているとは言えない。さらに，プログラムコードが漏えいした場合には，不正な識別コードを容易に作り出されてしまう問題もある。より高い安全性を確保するためには，利用者 ID など意味のあるデータを含まないセッション ID を用いてセッション管理を行う方がよいだろう。セッション ID はログイン時に生成することになるね。開発の期間は多少長くなるかもしれないが，セキュリティの方が重要だ。セッション管理の方式を見直してくれ。

G 君　：はい，分かりました。

設問１　セッション管理方式について，(1)，(2) に答えよ。

（1）表 1 中の [a] に入れる適切な字句を 10 字以内で答えよ。

（2）表 1 中の [b]，[c] に入れる適切な字句を，“ あり ”，“ なし ”のいずれかで答えよ。

設問２　セキュリティを考慮したコーディングについて，(1)，(2) に答えよ。

（1）本文中の下線①の処理は一般に何と呼ばれる処理か，10 字以内で答えよ。

（2）利用者からの入力データを含む Web ページを生成してブラウザに表示する場合の対策として，表 2 中の [d] ，[e] に入れる適切な文字列を答えよ。

設問３　G 君が設計した識別コードについて，(1)，(2) に答えよ。

（1）本文中の [f] ～ [i] に入れる正しい数値を答えよ。

（2）本文中の下線②で安全性が高くなっているとは言えないと F 主任が指摘した理由は何か。55 字以内で具体的に述べよ。ただし，プログラムコードは漏えいしていないものとする。

設問４　G 君は，F 主任が指示したセッション ID を，暗号を使わずに設計することにした。このセッション ID の設計において，不正なセッション ID を容易に作り出すことをできなくするために，セキュリティの面で留意すべきことを二つ挙げ，それぞれ 20 字以内で述べよ。

問２　情報資産に対する脅威への対策に関する次の記述を読んで，設問 1～3 に答えよ。

B 社は，従業員数 1,000 名の中堅の電子機器メーカである。汎用的な電子機器の設計，製造のほか，電子機器の受託開発も請け負っており，多品種少量生産が特徴となっている。

B 社の開発部では，紙書類の削減を目指し，設計及び開発に関連した電子文書（以下，設計開発文書という）の管理システム（以下，Y システムという）を開発することになった。Y システムの開発業務は，情報システムの開発，運用を主業務とする子会社の C 社に委託する。Y システムには，設計図面などの機密情報も格納するので，セキュリティを重視する必要がある。そこで，セキュリティ機能の設計を担当することになった C 社の H 主任は，まず，Y システムの要件，考慮事項の分析及び B 社における情報資産の管理規程の調査を実施した。

〔Y システムの要件と考慮事項〕

Y システムの要件の一部は，次のとおりである。

(1) サーバ上で動作する文書管理アプリケーション（以下，文書管理 AP という）を開発する。

(2) 文書管理 AP は，HTTP と HTTPS だけを利用して利用者のクライアント PC と通信することとし，Y システムの利用者が，ブラウザを利用して文書管理 AP にアクセスできるようにする。

(3) Y システムは，設計開発文書の種類ごとに異なるサーバで文書管理を行う構成とする。利用者からは，設計開発文書の種類ごとに異なる URL をもつサーバ群として見える。

(4) B 社内で認証局を運用し，B 社の全従業員に公開鍵証明書を配布するというプロジェクトが並行して進められている。この公開鍵証明書を利用したディジタル署名の機能を，Y システムに実装する。

Y システムの主な利用者は B 社開発部の従業員であり，各従業員に対して個別の利用者 ID が発行される。

B 社では，開発を担当する従業員数よりもはるかに多くの種類の電子機器を開発しているので，一人の従業員が担当する電子機器の種類が多岐にわたることも珍しくない。従業員ごとにアクセス可能な設計開発文書を指定することは可能であるが，多数の設計開発文書に対して，その都度，個別にアクセス権を設定することは，業務効率を著しく損なうと考えられる。設計開発文書に対するアクセス制御を検討する際には，この点を考慮する必要がある。

〔情報資産の管理規程〕

B 社では，情報資産の区分に応じた管理規程を情報セキュリティポリシで定めている。設計開発文書に対しては，次の管理規程が適用されることになる。

・当該情報資産の所管部門の部門長（以下，情報オーナという）が指定した者にだけ参照権限又は更新権限を与える。
・参照時の履歴として参照日時と参照者を特定できる情報を残す。また，更新時の履歴として更新日時と更新者を特定できる情報を残す。
・社外への送信時には，情報を暗号化する。
・社内への送信時及び情報システムや記録媒体への保存時の情報の暗号化については，情報オーナが指

定する。
・社外への送信時，社内への送信時，及び情報システムや記録媒体への保存時には，改ざんを検知できる手段を講じる。

設計開発文書の参照権限，更新権限及び暗号化については，設計開発文書の情報オーナであるB社開発部のJ部長の判断によって，次のような方針が決まっている。
・設計開発文書については，開発部の従業員に参照権限と更新権限を与える。他部署の従業員については，参照が必要となる頻度が低いので，必要なときに申請し，情報オーナが承認した場合に参照権限を与える。
・設計開発文書の送信時には，あて先が社内であっても暗号化する。情報システムへの保存時には，暗号化する必要はない。

〔設計開発文書の承認プロセス〕
B社における設計開発文書の承認プロセスは，次のとおりである。
（1）ある機器の設計開発文書の作成が一通り終了した時点で，当該機器の開発チームが当該設計開発文書の内容についてレビューを実施する。
（2）レビューが終了した時点で，開発チームのリーダ（以下，チームリーダという）が，当該設計開発文書を承認する。
（3）承認後の設計開発文書を更新する場合は，改めてチームリーダが承認する。ただし，チームリーダがレビューを必要と判断した場合は，承認に先立って開発チームでのレビューを実施する。

〔対策の選定〕
H主任は，Yシステムと設計開発文書に対する脅威のうち，技術的に対処が可能なものを抽出し，〔Yシステムの要件と考慮事項〕及び〔情報資産の管理規程〕の内容を考慮の上，対策を立案した。対策の一覧（一部）を表1に示す。

表１　対策の一覧（一部）

脅威	対策		
	抑止，予防	検出	回復
漏えい	A）表 2 に従って，各利用者 ID に参照権限を与える。 B）設計開発文書へのアクセスに失敗した場合，日時，利用者 ID，対象文書ファイル名，エラーコードを記録する。	－	－
改ざんと破壊	C）表 2 に従って，各利用者 ID に更新権限を与える。	D）設計開発文書の承認時，承認者が文書にディジタル署名を付与する。 E）設計開発文書が更新された場合，日時，利用者 ID，対象文書ファイル名を記録する。	F）バックアップからリストアする。
不正アクセス	G）不要なサービスを停止する。 H）セキュリティパッチを適用する。	I）文書管理 AP 及び OS のアクセスログを監視する。 J）［　a　］による監視を行う。	－
サービス不能攻撃	G）不要なサービスを停止する。 H）セキュリティパッチを適用する。	K）ネットワークトラフィックを監視する。 L）文書管理 AP 及び OS の［　b　］の稼働状況を監視する。	－
⋮	⋮	⋮	⋮

表２　参照権限及び更新権限の設定

権限／対象者	参照	更新
開発部の従業員	全設計開発文書について許可する。	開発を担当している機器の設計開発文書についてだけ許可する。
他部署の従業員	参照を承認された設計開発文書についてだけ許可する。	全設計開発文書について不許可とする。

〔レビューの実施と対応〕

H 主任が，表 1 の対策について B 社のレビューを受けたところ，次の指摘を受けた。

(1) ①対策 B）は，〔情報資産の管理規程〕の内容から考えて，不適切である。

(2) ②対策 C）は，〔Y システムの要件と考慮事項〕の内容から考えて，不適切である。

(3) 複数のサーバ上で記録されるログの整合性を保つため，各サーバ間で［　c　］が確実に実施されるようにする。

H 主任は，（1）～（3）の指摘に対して次のように対応した。
（1）対策 B）を変更し，設計開発文書への参照は，すべて記録することにした。
（2）表 2 を訂正し，開発部の従業員による更新には，参照時と同様の権限を与えた。
（3）ログ取得を行う対策 B），E）において，［　c　］の実施が必要である旨を明記した。

〔Y システムの要件の変更〕

H 主任が指摘事項への対応を進めているときに，Y システムの要件（4）における認証局の利用開始が，種々の理由によって延期されることが決定した。Y システムの稼働開始時に公開鍵証明書を利用できなくなったので，B 社開発部と C 社で検討した結果，公開鍵証明書の利用については，Y システム設計時の考慮対象外とすることに決まった。

レビューへの対応と Y システムの要件の変更に伴う表 1 及び表 2 の修正後，再レビューが実施された。対策 D）の変更によって，承認プロセスをシステム的に支援できるようにすることを条件に表 1 及び表 2 は承認された。H 主任は，要件の変更によって利用を見送ることになったディジタル署名の代わりに［　d　］による③設計開発文書のハッシュ値を登録管理するためのシステムを開発した。

設問１　表 1 及び本文中の［　a　］～［　d　］に入れる適切な字句を解答群の中から選び，記号で答えよ。

解答群

ア　AES	イ　IDS	ウ　SHA-256
エ　SNMP	オ　アクセス制御	カ　時刻同期
キ　脆弱性検査	ク　プロセス	ケ　変更管理

設問２　B 社のレビューでの指摘事項について，(1)，(2) に答えよ。
（1）本文中の下線①が指摘された理由を，35 字以内で述べよ。
（2）本文中の下線②が指摘された理由を，35 字以内で述べよ。

設問３　本文中の下線③のシステムが具備すべき機能を，40 字以内で述べよ。また，そのシステムにハッシュ値を登録する際の運用上の条件を，30 字以内で述べよ。

問３　IC カードを利用したりん議システムに関する次の記述を読んで，設問 1～4 に答えよ。

D 社は，従業員数 2,000 名の中堅電機メーカである。D 社では，業務改革のために，グループウェアを利用したりん議システムパッケージを採用し，りん議システムを構築することにした。これまでは，りん議書の起案や承認には印鑑を用いていたが，りん議システムでは，印鑑の代わりに電子的な手段を採用する必要がある。

SI ベンダの K 氏がこのりん議システムの開発を担当することになり，りん議システムパッケージに付加する機能の具体的な要件について，人事総務部の L 氏と打ち合わせた。

〔ディジタル署名の仕組み〕

K 氏は，ディジタル署名の仕組みを L 氏に説明し，ディジタル署名機能を組み込み，ISO/IEC 7816 に準拠した IC カードの使用を提案した。次は，そのときの K 氏と L 氏の会話である。

L 氏：なるほど。鍵ペアを作成し，これを IC カードに格納して従業員に持たせればよいということですね。

K 氏：鍵ペアを作成し，IC カードに格納するだけでは安心できません。第三者信頼機関である［　a　］が必要です。

L 氏：それはなぜですか。

K 氏：ディジタル署名を検証するとき，検証に使う公開鍵がだれのものであるかを確認しなくてはなりません。このために公開鍵証明書を使います。公開鍵証明書には，所有者の公開鍵や名前，有効期限などの情報が格納され，［　a　］がディジタル署名を付与します。

L 氏：なるほど。そのディジタル署名を検証することによって，公開鍵証明書の内容が［　b　］されていないかを判断できますね。

K 氏：はい。さらに，IC カードをディジタル署名に使うためには，秘密鍵の保管方法に留意することが必要です。

K 氏は，社内に［　a　］を設置し，IC カード内で秘密鍵を生成した後，公開鍵証明書を IC カードに内蔵し，従業員証とすることを L 氏に提案し，了承を得た。

〔IC カードに必要な PIN の設定と運用〕

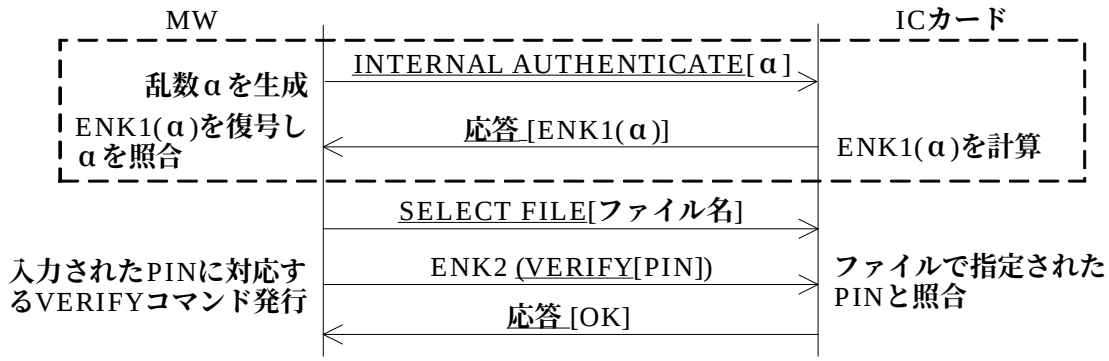
L 氏と K 氏は，各従業員が IC カードを利用する場合に必要となる PIN（Personal Identification Number）の扱いを検討した。IC カードには，新規発行時に初期 PIN が設定され，従業員には IC カード受領後に初期 PIN を変更することを義務付けた。また，PIN の設定に関するガイドラインを図 1 のとおり提示することにした。

1. アルファベットの大文字と小文字，数字，記号をランダムに並べ，当該本人情報から推測できるような情報を含めない。
2. 一般に使用される単語，及びテレビ，ラジオなどのメディアで使用されている流行語や時事用語などは使用しない。
3. 8 文字以上の文字列とする。

図１　ＰＩＮの設定に関するガイドライン

従業員が PIN を入力し，連続して 5 回照合に失敗すると，その IC カードを使用不可能な状態（以下，ロック状態という）とする。従業員が PIN を忘れてしまった場合や，ロック状態となった場合は，人事総務部が管理者として IC カードを回収し，ロック状態を解除した後，再度初期 PIN を設定して，従業員に返却する。従業員が退職した場合は，IC カードを回収するとともに，退職日に公開鍵証明書の失効処理を行う。

りん議システムのクライアント PC にはリーダライタ（以下，R/W という）が接続されており，PC では，R/W に挿入された IC カードにアクセスするミドルウェア（以下，MW という）が動作する。K 氏は，IC カードを R/W に挿入した場合の，MW と IC カード間のプロトコルを図 2 のとおり L 氏に示し，①PIN に対するセキュアメッセージングの内容を説明した。



注　下線はコマンド名，[ ]内は主な引数
ENK1(α)：ICカードとMWが共有している暗号化鍵によって，αを暗号化
ENK2(VERIFY[PIN])：セキュアメッセージ化されたVERIFYコマンド

図２　MWとＩＣカード間のプロトコル

〔IC カードのファイルシステム設計〕

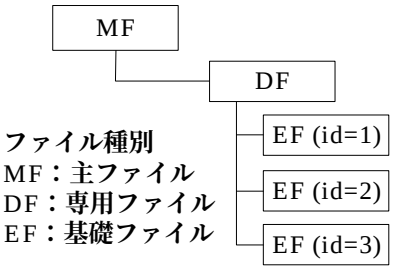
次に，K 氏は IC カード内のファイルシステムを設計した。各ファイルには，表 1 に示すセキュリティ属性を複数設定することができる。

表１　セキュリティ属性

セキュリティ属性	説明
RW	設定されたファイル内のデータを IC カード外から読み書き可能
XX	設定されたファイル内のデータを IC カード外から読み書き不可能
R	設定されたファイル内のデータを IC カード外から読出し可能
W	設定されたファイル内のデータに IC カード外から書込み可能
LN=n	PIN 照合の許容失敗回数で，連続して n 回照合に失敗するとロック状態となる。
L	設定されたファイル内のデータとの照合によってロック状態を解除可能

K 氏は，図 3 に示すようにファイルシステムを構成し，各ファイルのセキュリティ属性を設定した。

〔ファイルシステムの構成〕



〔セキュリティ属性の設定〕

ファイル種別（識別子）	格納情報	セキュリティ属性
EF（id=1）	PIN	c d
EF（id=2）	所有者の秘密鍵	e
EF（id=3）	所有者の公開鍵証明書	f

図３　ファイルシステムの構成とセキュリティ属性の設定

〔りん議システムパッケージへのプログラムの組込み〕

K 氏は，想定するりん議システムパッケージに，承認及びその確認処理を組み込むことを検討した。りん議システムには，承認時にディジタル署名を要求するプログラムと，承認を確認するときにディジタル署名を検証するプログラムが必要である。また，ディジタル署名付きりん議書には，IC カードによって生成された署名値と IC カードから読み出した公開鍵証明書を含めることにした。K 氏は，MW が提供する API（Application Program Interface）を用いて開発したこれらのプログラムをりん議システムパッケージに組み込み，りん議システムを完成させた。

設問１　本文中の [a], [b] に入れる適切な字句を答えよ。

設問２　りん議システムに組み込むプログラムにおいて，ディジタル署名を検証するために必要となる承認者の公開鍵証明書の有効性確認に必要な情報を二つ答えよ。

設問３　図 3 中の [c] ～ [f] に入れるセキュリティ属性を，表 1 から選んで答えよ。

設問４　IC カードの PIN について，(1)～(4) に答えよ。

（1）図 1 中の 2.で想定している攻撃手法の名称を答えよ。

（2）図 2 中の破線で囲まれた部分の手順の目的を，15 字以内で述べよ。

（3）本文中の下線①が想定する脅威を 10 字以内，その想定する発生場所の範囲を 15 字以内で述べよ。

（4）IC カードにおける PIN の運用条件を満たすためには，図 3 のファイルシステムに不足しているファイルが一つある。ファイルの用途を 20 字以内，格納情報を 10 字以内で述べよ。

問４　リモートアクセスシステムの構築に関する次の記述を読んで，設問 1～3 に答えよ。

E 社は，従業員数 120 名のソフトウェア開発業者で，海外にも開発拠点をもっている。従業員は，E 社のメールサーバを利用した電子メール（以下，メールという）の送受信を行っている。E 社の営業部には 20 名の従業員が在籍しており，国内外の出張がしばしば発生する。しかし，出張時にメールの送受信ができず，これまでにも営業部の従業員から苦情が寄せられていた。そこで，E 社の情報システム部では，リモートアクセスシステムを新たに構築することになり，セキュリティ技術に詳しい M 君が中心となって製品の検討を開始した。

〔リモートアクセスシステムと製品の概要〕

リモートアクセスシステムは，サーバ（以下，SV という）とクライアント（以下，CL という）から構成される。リモートアクセスシステムの導入によって，従業員は社外のネットワークから，SV を経由してメールサーバに接続することが可能になる。M 君は，リモートアクセスシステムを構築するに当たり，市販製品の中から製品 P と製品 Q を候補として選び出した。製品 P は，Web インタフェースをもち，クライアントソフトウェアとしてブラウザを使用する。利用者の認証には，利用者 ID とパスワードを利用する。一方，製品 Q は，サーバソフトウェア及びクライアントソフトウェアから構成された製品で，利用者の認証には，各従業員が作成した公開鍵と秘密鍵のペアを利用する。

〔製品仕様の確認〕

M 君は，情報システム部主任の N 氏と共同で，製品 P，Q の仕様確認を行うことにした。次は，そのときの M 君と N 氏の会話である。

N 氏：リモートアクセスシステムは，主に出張先や外出先からメールを送受信するときに使用することになります。インターネット経由の接続では，不正な攻撃者が従業員になりすますことや通信路での［　a　］が問題になりますね。

M 君：はい。SV に接続してくる利用者の認証と通信路の暗号化は必須の機能です。

N 氏：まず，製品 P の処理手順を教えてください。

M 君：はい。製品 P はサーバ認証に SSL を使用します。図 1 に製品 P の処理手順を示します。なお，SSL ハンドシェイク手順は簡略化してあります。

MAC：メッセージ認証符号

図１　製品Ｐの処理手順

M 君：CL から接続要求があると，SV から CL に対して SV の証明書が送られます。CL は SV の証明書の有効性確認を行った後，［ b ］の公開鍵で暗号化した乱数を SV に送信します。また，この乱数を基に，共有鍵が生成されます。次に，SV は，共有鍵を用いて，それまでに送受信されたすべてのメッセージの MAC 値を計算し，CL に返信します。そして，CL は，MAC 値の検証を行います。

N 氏：製品 P では，利用者の認証をどのように行うのですか。

M 君：利用者がブラウザに入力した利用者 ID とパスワードを確認することによって，認証を行います。送受信される利用者 ID とパスワードは，共有鍵によって暗号化されています。また，利用者 ID とパスワードは，従来から従業員に配布しているものを利用できます。

N 氏：製品 P は，専用クライアントソフトウェアをインストールする必要がないので，比較的容易に導入できますね。

M 君：はい。しかし，クライアント端末で使用できるのはブラウザに限定されるので，既存のメールシステムの改修が必要になります。

N 氏：そうですか。次に，製品 Q の仕様について確認しましょう。

M 君：はい。図 2 に製品 Q の処理手順を示します。

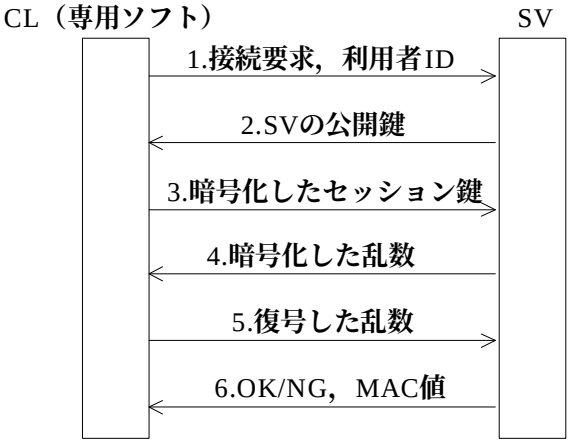


図２　製品Qの処理手順

N 氏：製品 Q では，あらかじめ設定しておいたアドレスの SV に接続し，認証処理を行うのですね。図 2 の処理手順 2 で，SV が公開鍵を送信していますが，製品 Q では，SV の公開鍵を CL に事前登録するようになっています。それはなぜですか。

M 君：セッション鍵は，図 2 の処理手順 2 の後に生成されますが，SV の公開鍵を事前登録していない場合，不正な攻撃者がセッション鍵を入手できる可能性があるからです。製品 Q では，その対策として，送信されてきた SV の公開鍵が，事前登録されている公開鍵と一致するかどうかを照合します。

N 氏：利用者の認証はどのように行うのですか。

M 君：SV は，図 2 の処理手順 4 で送信した乱数の値と，処理手順 5 で CL が返信した乱数の値が同じかどうかを照合し，正当な利用者かどうかを確認します。

〔製品の運用確認〕

N 氏：製品の仕様は分かりました。製品を選ぶに当たって，運用面についても確認しておきたいのですが，どのような検討課題がありますか。

M 君：まず製品 P ですが，正しい SV に接続していることを保証するためには，SV の証明書の有効性確認だけでは十分ではありません。（ア）接続している URL の確認や，（イ）SV の証明書に関するブラウザの警告メッセージの確認を，利用者に促す必要があります。

N 氏：従業員の中には，警告メッセージを読み飛ばしてしまう人も多いようですね。製品 Q についても，何か検討課題はありますか。

M 君：そうですね。鍵の配布方法や登録方法を検討する必要があります。まず，社内サイトを活用して，SV の公開鍵を安全に事前配布します。また，各従業員が専用のソフトウェアを利用して秘密鍵と公開鍵のペアを作成しますが，作成した公開鍵を安全な方法で SV に登録しなければなりません。

N 氏：それでは，製品 P，製品 Q の利点と欠点を整理しましょう。

M 君：製品 P は，専用クライアントソフトウェアを必要としないことから，クライアント端末に導入しやすいと思います。しかし，既存のメールシステムの改修が必要なので，製品 Q に比べて費用が

かかります。製品 Q は，製品 P と比べて安全に運用される可能性が高いと思います。しかし，運用においては，従業員の公開鍵の登録や更新をサポートしなければなりません。

N 氏：リモートアクセスシステムの利用者が少ないことを考慮すると，製品 Q の方がよいのではないでしょうか。

M 君：それでは，製品 Q を購入して，リモートアクセスシステムを構築することにします。

設問１　本文中の［　a　］，［　b　］に入れる適切な字句を，それぞれ 5 字以内で答えよ。

設問２　製品 P について，(1)～(3) に答えよ。

（1）図 1 の処理手順 2 と 3 の間で行われる SV の証明書の有効性確認は，三つの処理からなる。三つの処理のうち，署名値の照合，証明書の有効期限確認のほかに，証明書の有効性確認の観点から行うべき処理を，15 字以内で述べよ。

（2）図 1 の処理手順 3 と 4 では，攻撃者が SV になりすますことができないような処理手順になっている。なりすましができない理由を，25 字以内で述べよ。

（3）本文中の下線（ア）及び（イ）について，利用者が確認を怠った場合に攻撃を許すことになってしまう理由を，それぞれ 30 字以内で述べよ。

設問３　製品 Q について，(1)，(2) に答えよ。

（1）図 2 の処理手順で，従業員の公開鍵を用いて生成されるデータを，番号 1～6 の中から選んで答えよ。

（2）SV の公開鍵を事前登録せず，図 2 の処理手順において公開鍵の照合処理を行わなかった場合に，攻撃者がセッション鍵を入手できる可能性がある。可能性として考えられる入手方法を二つ挙げ，それぞれ 40 字以内で具体的に述べよ。

プログラム言語 Perl の用例・解説

Perl を使用した問題では，各問題文中に注記がない限り，次に示す用例に従って記述する。
なお，用例は，解答で使用する演算子，関数，予約語などを制限するものではない。

種類	用例 解説

1.注釈

#	#ここにコメントを書く 行末までが注釈となる。

2.リテラル

スカラ	123 10 進数 123 である。
	12.3 10 進数 12.3 である。
	4E-5 10 進数 4×10^{-5} である。
	0x9f 16 進数 9F である。
	0147 8 進数 147 である。
	0b010111 2 進数 010111 である。
	$var = "hello"; print '$var ', "$var ",`echo world`; 変数 var に文字列“ hello ”を代入する。文字列のスカラ'$var ', "$var ", `echo world` を出力する。"$var "は変数を展開し，`echo world`はコマンドの出力を展開するので，出力は“ $var hello world ”となる。
	＼n 制御文字（改行）である。
	＼r 制御文字（復帰）である。
	＼t 制御文字（水平タブ）である。

リストリテラル	('a', 'b', 'c')
	リスト('a', 'b', 'c')である。
	('a', 'b', 'c')[0]
	リスト('a', 'b', 'c')の１番目の要素'a'である。
	()
	空リストである。
	('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie')
	キーa，b，c に，それぞれ値 alpha，bravo，charlie を結び付けたハッシュである。
ファイルハンドル	STDIN
	標準入力である。
	STDOUT
	標準出力である。
	STDERR
	標準エラー出力である。
	AV
	コマンドラインから指定されたファイル名のリストを順に読み込むためのファイルハンドルである。

3.変数

スカラ変数	$var
	スカラ変数 var である。
配列変数	@ary
	配列変数 ary である。
配列要素	$ary[6]
	配列変数 ary の 7 番目の要素である。
ハッシュ変数	%hash
	ハッシュ変数 hash である。
ハッシュ要素	$hash{'a'}
	ハッシュ変数 hash の要素のうち，キーa に結び付けられた値である。
局所的な変数	{my $var;}
	{}内を有効範囲とする変数 var の宣言である。
$_	$_ = "abc"; if (/b/) print "match";
	パターンマッチの演算子が省略されたとき，$_の文字列“ abc ”が//内のパターン b と一致するかどうかを判定し，“ match ”が出力される。
@ARGV	@ARGV
	コマンドライン引数のリストを格納する配列変数である。
@_	@_
	サブルーチンに渡す引数のリストを格納する配列変数である。

4.演算子

->	$object->method1
	オブジェクト object のメソッド method1 を呼び出す。
	Class->method2
	クラス Class のメソッド method2 を呼び出す。
++, --	$a++
	変数 a を評価した後に 1 を加算する。
	--$b
	変数 b から 1 を減算した後に評価する。
!, +（単項), -（単項）	!$a
	変数 a の論理否定である。
	+123
	正の数 123 である。
	-123
	負の数 123 である。
=~, !~	$html_contents =~ /<b>/
	変数 html_contents の値に，文字列“ <b> ”が含まれているときに真を返す。
	$html_contents !~ / /
	変数 html_contents の値に，文字列“ ”が含まれていないときに真を返す。
*, /, %	314 * 34
	314 と 34 の乗算である。
	6 / 469
	6 を 469 で割る除算である。
	34 % 6
	34 を 6 で割る剰余演算である。
+, -, .	3.14 + 2.72
	3.14 と 2.72 の加算である。
	220-8125
	220 から 8125 を引く減算である。
	"IPA"."JITEC"
	文字列“ IPA ”と“ JITEC ”の連結である。
<, >, <=, >=, lt, gt, le, ge	1 < 2
	数値 1 と 2 を比較し，演算子の左側が右側より小さいので真を返す。数値の関係演算子には，ほかに>，<=，>=がある。
	"b" lt "a"
	文字列“ b ”と“ a ” を比較し，演算子の左側が右側より小さくないので偽を返す。文字列の関係演算子には，ほかに gt，le，ge がある。
==, !=, <=>, eq, ne, cmp	1 <=> 2
	数値 1 と 2 を比較し，演算子の左側が右側より大きければ 1，等しければ 0，小さければ－1 を返すので, この場合は－1 を返す。数値の比較演算子には, ほかに==, != がある。
	"b" cmp "a"

	文字列“b”と“a”を比較し，演算子の左側が右側より大きければ1，等しければ0，小さければ−1 を返すので，この場合は 1 を返す。文字列の比較演算子には，ほかに eq，ne がある。
&&	`$x >= 0 && $x < 10` 変数 x の値が 0 以上かつ 10 未満なら真を返す。
||	`$x < 0 || $x >= 10` 変数 x の値が 0 未満又は 10 以上なら真を返す。
..	`@card = (1 .. 52)` 1 から 52 までの連続する整数を配列変数 card に代入する。
=，+=，-=， *=，/=，%=	`$a = 1` 変数 a に 1 を代入する。
	`$a += 10` 変数 a の値に 10 を加算して a に代入する。 代入演算子には，ほかに-=，*=，/=，%=がある。
=>，，	`%hash = ('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie')` a に alpha，b に bravo，c に charlie を結び付けたハッシュをハッシュ変数 hash に代入する。
not	`not $a` 変数 a の論理否定である。
and	`$a < 0 and $b == 0` 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の論理積である。
or， xor	`$a < 0 or $b == 0` 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の論理和である。
	`$a < 0 xor $b == 0` 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の排他的論理和である。

注　演算の優先順位は，上表の枠の順である。

5.文

if	`if ($var == 1) {`
	`  print "a";`
	`} elsif ($var == 2) {`
	`  print "b";`
	`} else {`
	`  print "c";`
	`}`
	変数 var の値が 1 なら“a”を，2 なら“b”を，それ以外なら“c”を出力する。

while	$i = 1;
	while($i <= 10) {
	　print $i++, "\n";
	}

	変数 i の値を 1 から 1 ずつ増やし，10 回出力する。

for	for($i = 1; $i <= 10; $i++){
	　print "$i\n"1;
	}

	変数 i の値を 1 から 1 ずつ増やし，10 回出力する。

foreach	foreach $i (1, 3, 5){
	　print "$i\n";
	}

	変数 i にリストの各要素 1，3，5 を順に代入し，3 回出力する。

next	for ($i = 1; $i <= 10; $i++) {
	　next if $i % 2;
	　print "$i\n";
	}

	変数 i が 2 で割り切れないとき，ループ本体の next 行より後を実行しないので，偶数を出力する。

6.正規表現

\	/\. \^\$\[|\+*\?\{\(\)\/\\/
	次の 1 文字そのものを表す。“ .^$[|+*?{()/ \ ”と一致する。
.	/www.ipa.go.jp/
	改行文字以外の任意の 1 文字と一致する。“ wwwdipa,go@jp ”と一致する。
^	/^ab/
	先頭が“ ab ”である文字列と一致する。“ abc ”と一致するが，“ cab ”とは一致しない。
$	/yz$/
	末尾が"yz"である文字列と一致する。“ xyz ”と一致するが，” yza ”とは一致しない。
+	/qo+d/
	直前の 1 文字 o の 1 回以上の繰返しと一致する。“ god ”や“ gooood ”と一致するが，“ gd ”とは一致しない。
*	/go*d/
	直前の 1 文字 o の 0 回以上の繰返しと一致する。“ gd ”，“ god ”や“ gooood ”と一致する。
?	/colou?r/
	直前の 1 文字 u の 0 回又は 1 回の出現と一致する。“ color ”又は“ colour ”と一致する。

{m}, {m,n}	/co{2}1/
	直前の 1 文字 o の 2 回の繰返しと一致する。“cool”と一致するが,“col”や“coool”とは一致しない。
	/go{1, 3}d/
	直前の 1 文字 o の 1～3 回の繰返しと一致する。“qod”や“qood”と一致するが,“gd”や“gooood”とは一致しない。
(…)	/<(h.)>/
	()内の文字列と一致するパターンを部分パターンとしてまとめる。“<h1>”と一致した場合は“h1”が,“<hr>”と一致した場合は“hr”が，まとめられる。
＼1, ＼2, …	/<(..)><([bp])>JITEC<＼/＼2><＼/＼1>/
	左から順に()内のパターンと一致した文字列が＼1, ＼2, …に割り当てられる。“ JITEC ”と一致するが,“<td>JITEC</P></td>”とは一致しない。
[…]	/<h[12r]>/
	[]内で指定した文字 1，2 又は r のどれか一つと一致する。“<h1>”,“<hr>”と一致するが,“<h3>”や“<HR>”とは一致しない。
	/[^0-9]/
	[]内で指定した 0～9 以外の 1 文字と一致する。“a”と一致するが,“3”とは一致しない。
…|…	/<(a href|img src)=/
	|で区切られた“a href”又は“img src”のどちらか一方と一致する。“<a href=”や“<img src=”と一致するが,“<A HREF=”や“<img height=”とは一致しない。

7.サブルーチン

定義	sub greeting{
	　print "hello Perl＼n";
	}

	“hello Perl”を出力するサブルーチン greeting を定義する。
呼出し	subroutine ($arg1, $arg2);
	サブルーチン subroutine を引数 arg1 と arg2 で呼び出す。()を省略して“subroutine $arg1, $arg2; ”とする表記もある。
戻り	return －1;
	サブルーチンから抜け出し，値－1 を返す。

8.モジュール

use	use CGI;
	モジュール CGI を 1 度だけ読み込み，利用可能にする。

9.メソッド呼出し

->	`$object->method1(arg1);`
	演算子 ->を使って，オブジェクト object のメソッド method1 を引数 arg1 で実行する。
	`Class->method2(arg1, arg2);`
	演算子->を使って，クラス Class のメソッド method2 を引数 arg1 及び arg2 で実行する。

10.文字列操作関数

chomp	`chomp @lines;`
	配列変数 lines の各要素の末尾にある改行文字を削除する。
eval	`eval $exp_str;`
	変数 exp_str の内容を Perl プログラムとして解釈し実行する。
length	`length $long_str;`
	変数 long_str に格納される文字列の文字数を返す。

11.配列・ハッシュ操作関数

keys	`%hash = ('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie');`
	`foreach $key (keys %hash){`
	`  print "$key＼n";`
	`}`

	ハッシュ変数 hash のキーのリストを取り出し，各キーを出力する。この場合は，“a”,“b”,“c”を順不同に出力する
shift	`$next = shift @queue;`
	配列変数 queue の先頭要素を取り除いて詰め，取り除いた値を変数 next に代入する。
sort	`@pile = sort @jumble;`
	配列変数 jumble の値を文字列の大小比較によって昇順に整列し，配列変数 pile に代入する。
	`@pile = sort {$b <=> $a} @jumble;`
	配列変数 jumble の値を数値の大小比較に従って降順に整列し，配列変数 pile に代入する。
split	`@fields = split ',',$csv;`
	変数 csv の値をコンマで区切って分割したリストを配列変数 fields に代入する。

12.検索・置換関数

M/…/又は /…/	$html_contents =~ /<b>/i; 変数 html_contents の値が，文字列“<b>”又は“<B>”を含んでいるかどうかを判定する。i は，大文字，小文字の区別をしないオプションである。
S/…/…/	$html_contents =~ s/ /＼n/gi; 変数 html_contents の中の文字列“ ”，“ ”，“ ”又は“ ”を改行文字に置換する。g は，一致したすべての文字列を置換するオプションである。
$`, $&, $', $1, $2, …	'The date is 1970-01-23.' =~ /([0-9]{4})-([0-9]{2})-([0-9]{2})/; print "String before the date: $`＼n"; print "Date: $&＼n"; print "String after the date: $'＼n"; print "Year: $1＼n", "Month: $2＼n", "Day: $3＼n"; 文字列“The date is 1970-01-23. ”に対して，一致した部分の前の文字列，一致した文字列，一致した部分の後ろの文字列をそれぞれ変数 `, &, ' に代入する。また，()で囲まれた部分パターンと一致した文字列を，1 番目から順に変数 1，2，3 に代入する。これらを利用し，“String before the date:The date is”，“ Date:1970-01-23 ”，“ String after the date:. ”，“ Year:1970 ”，“ Month:01 ”，“ Day:23 ”の 6 行を出力する。

13.入出力操作関数

open	open LOG, '>>cgi.log'; ファイル cgi.log を追記モードで開き，ファイルハンドル LOG に対応付ける。
<*filehandle*>	$line = <USER_FILE>; ファイルハンドル USER_FILE から 1 行を読み込んで変数 line に代入する。
<>	@records = <>; 標準入力（コマンドライン引数があるときは，コマンドライン引数で指定されたファイル）から順にデータを読み込み，すべての行を配列変数 records に代入する。
print	Print LOG "sync.＼n"; ファイルハンドル LOG に対応するファイルに文字列を出力する。
close	close LOG; ファイルハンドル LOG に対応するファイルを閉じる。

14.システムインタフェース

die	open(FILE, 'a_file')or die 'cannot open a_file'; ファイル a_file を開く。開くのに失敗したとき，“cannot open a_file”というメッセージを出力して実行を終了する。
system	system 'a.out'; コマンド a.out を実行し，コマンドが終了するまで待機する。