今回はセキュリティについてですが、特にアクセスの権限を制御してデータを保護することについて学びます。
教科書では、第13章　セキュリティと信頼性になります。

OSの目的は、資源の仮想化、効率的管理、そして保護・セキュリティの確保です。
保護は極めて重要な機能であり、近年はインターネットの発展により社会基盤の視点から必要不可欠な機能となっています。

オペレーティングシステム
（2024年 第11回）

アクセス制御とデータ保護、セキュリティについて

前回の課題について

詳細は「第10回の課題について.pdf」をみてください

（1）通信プロトコルは、物理的なものから、メッセージの内容、通信手順、誤りが生じたときの制御方法などの規定を階層で定めているものですので、これに従って処理することでメーカーが違った製品であっても、またOSが違っていても通信が行えます。したがって「エ」が適切であると考えられます。

（2）多数のパケットが多重化されて伝送路やルータを共有することで、たとえば混み合ったルートを避けることができるなど伝送路やルータの利用効率を上げることができる。

パケットという分割されたデータが送信中に途中で消失しても、その部分だけを送信しなおすだけで済み、エラー処理が行いやすい。

などが考えられます。

- コンピュータシステムの安全性を脅かす要因
  - 天災、人災
  - 建物、コンピュータ室への不法侵入
  - 停電、ハードウェア障害
  - ソフトウェア不良
  - 操作ミス
  - ネットワークでの盗聴、不法アクセス
  - 不正プログラム（ウィルスなど）
  - 不正なネットワークトラフィック
  - 非人為的なもの
  - 人為的なもの
    - 不注意によるもの
    - 故意（悪意）によるもの

コンピュータウィルスなどからコンピュータシステムを保護することを考えますが、まず、どのような脅威があるのか見てみますと、災害の類や何らかの障害、ミス、バグ、不正といったものがあり、また人為的になされたものかそうでないか、人為的なものでは不注意なのか悪意なのかなどの観点があります。

ここでは、バグによる予期しない誤った操作と、意図的な改ざんなど悪意ある操作を明確には区別しません。結果的にコンピュータシステムとその利用について影響を及ぼすことになるからです。

セキュリティ

- コンピュータセキュリティ
  - コンピュータシステムを災害、誤用および不正な利用から守ることであり、ハードウェア、ソフトウェア、データのいずれについてもその機密性、完全性、可用性を維持すること（wikipedia）
  - コンピュータに依存でき、ソフトウェアが期待どおりに動作するとき、そのコンピュータは安全である（UNIXセキュリティ：アスキー出版局）
- 講義では、ファイルに関する保護の概略の話に限定
  データの保護
  - 二次記憶装置 … ファイル管理に関する機構
  - メモリ　　… メモリ管理に関する機構
  - ネットワーク

コンピュータセキュリティの定義にはいくつかありますが、達成されるべき目標は、機密性・完全性・可用性を維持することになります。

先に見たように要因が様々ですので、何に関して取り組むかという点についても様々になります。
OSのレベルで資源（ハードウェアやプログラム、データ）を保護することを考え、講義ではファイルの保護に関する機構に関して、を扱います。

オペレーティングシステムにおける保護

マルチプロセスの環境下で、保護はプログラムの信頼性の観点から極めて重要な機能

- 保護の目的と目標
  - 機密性：許可された者が許可された方法でのみ情報にアクセスできる（守秘性）ことを確実にすること
  - 完全性：情報が破壊、改ざんまたは消去されていない状態を確保すること
  - 可用性：許可された者が、必要な時に中断することなく情報にアクセスできる状態を確保すること
    - エラー回復、バックアップ、多重化
  - 検証可能性：後で事象を客観的に分析できること

独立性や安全性確保のため、特定の資源と特定のユーザとを対応付けたい

- 個人や特定のグループのデータ
- 管理用データ、ハードウェア

保護の目的と目標について、改めて項目を挙げます。

保護のために、特定の資源とユーザを対応付けるようにします。ユーザの他にユーザのグループといった考えを導入します。

- オペレーティングシステムにおける資源の保護
  - 保護する資源 … ハードウェア、プログラム、情報（データ）
  - アクセス制御
    プログラムが資源にアクセスするとき、そのアクセスが正当なものであることを保証すること（許可されたプログラムに対して、操作を許可する）
    操作 … 生成、削除、読み書き、実行
  - 認証
    正当性を検証する作業
    コンピュータを利用しようとしているユーザにその権利があるかどうか、本人かどうかなどを確認すること（ユーザ認証）
    - ID（識別子）を対象とする
    - カーネル外で行われる

オペレーティングシステムが提供する最も基本的な要件はアクセス制御
- ユーザ（主体）に資源（対象）へのアクセス権（何をできるか）を設定する
- 「主体が誰であるか」を保証するために認証が重要となる

保護のやり方は、保護の対象である資源（ハードウェア、プログラム、データ）に対して、誰が、どのようにアクセスすることが正当であるかを確認することで行います。

どのようにアクセスできるかが、アクセス制御です。「生成」や「読み書き」などの操作を行うときに、それが許可されたものには正当であることを保証し、許可されていなければ拒否することです。
「誰が」、にあたるものがユーザの認証で、利用しようとしている人が権利をもっているのか、すなわち本人であるかを確認することです。

オペレーティングシステムが提供する最も基本的な保護の要件はアクセス制御で、これが成り立つために、「誰が」を保証する認証が重要になります。

アクセス制御

- アクセス制御の基本的な考え方
  アクセスを行う「主体」がアクセスされる「対象」に対してどのような操作が行えるかを規定する
  - 主体 … ユーザ、プロセス、プログラム、ハードウェア装置 など
    主としてユーザとプロセス
  - 対象 … ファイル、入出力装置などすべての資源、主体も
- 右記「アクセス制御行列」を実現する
  - そのまま表にするとサイズが非常に大きく、まばら
  - 追加・削除・変更が困難
  - アクセス制御リスト
  - ケーパビリティ(資格)リスト

  として実現

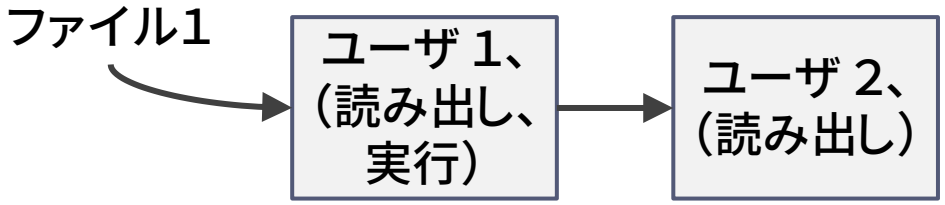
主体＼対象	ファイル1	ファイル2	プロセス1	
ユーザ１	読み出し 実行			
ユーザ２		読み出し 書き込み	中断	
ユーザ３	読み出し			

アクセス制御の基本的な考えは、アクセスを行う主体(ユーザ)が、アクセスの対象(ファイルなど)に対して、どの操作ができるかを規定することです。
たとえば、ユーザ1は、ファイル1に対して「読み出し」と「実行」の権限を持っているなどといった具合です。

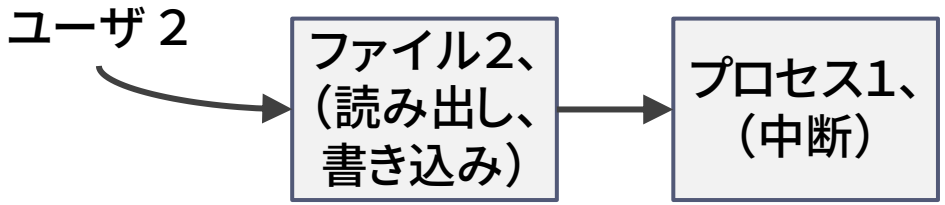
このような情報を管理するには、この図のような、主体と対象の資源を行・列に持つアクセス制御行列を作ればよいのですが、ユーザ数も資源数も多く、そのすべてのものを扱う(どのユーザもほとんどすべての資源に対してなんらかの権限をもつ)わけではないので、巨大でまばらな行列となって効率がよくありません。また、行や列の位置が固定されるので、追加や削除が面倒になります。

そこで、この中の情報をリストで持つことが考えられます。

- **アクセス制御リスト（ACL）**
  アクセス制御行列の各列に属する各々のセルの情報をリストとして管理する方式
  - ファイルのアクセス制御などでよく用いられる



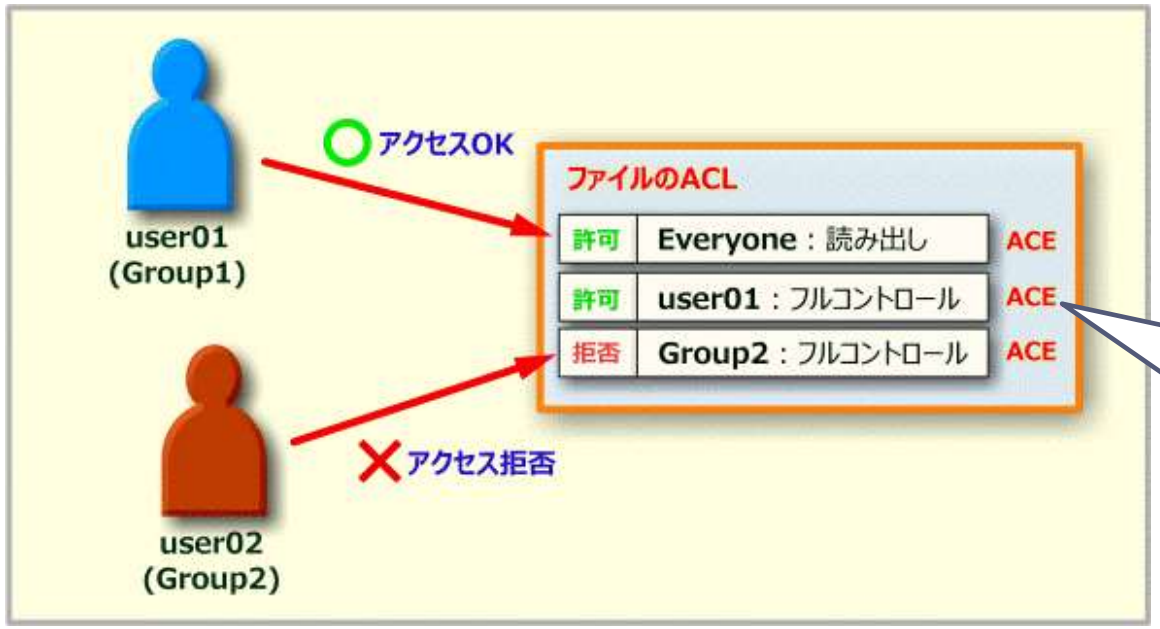
- **ケーパビリティリスト**
  ACLとは反対に、行に属する各々のセルをリストとして管理する方式



先の行列を列ごとにリストにしてもつ、すなわち、ファイルなど1つの資源から見て、それに対して、複数の主体がユーザ1は実行権限、ユーザ2は読み出し権限があるといったように権限をリスト要素としてつなぐものです。アクセス制御リスト（ACL：Access Control List）と呼ばれます。

それとは反対に、行ごとにリストにする、すなわち、1つのユーザからみて、そのユーザが複数の主体に対して、ファイル2には読み書きの権限、プロセス1には中断できる権限があるといった要素をつなぐものです。ケーパビリティリストと呼ばれます。

WindowsでのACLによるアクセス制御



ACE：アクセス制御エントリ
アクセスを許可するユーザやグループを記述したエントリ

- アクセスを許可するユーザやグループを明示的に列挙し、データにアクセスできるユーザを限定することが可能
（列挙されていないユーザやグループからのアクセスは許可されない）
- WindowsのACLには「許可」だけでなく、「拒否」というエントリもあり、アクセスを排除したい主体を明示的に指定できる

図は @IT> Windows Server Insider>Windows OS入門：第7回 より

WindowsのACLは、名前をご存知の方もいることと思います。

この図では「ファイル」に対して、<すべてのユーザが読み出し>、<user01がすべての権限>、<Group2がすべて拒否>となっています。

UNIXでのアクセス権

- UNIX系オペレーティングシステムでは、ファイルのアクセス権は
  所有者/グループ/第三者 の3つに対する 読み出し/書き込み/実行 の
  許可をそれぞれ指定する

  ファイル所有者以外をグループもしくは第三者のどちらかに分類
  - 自分が所有しているファイルやディレクトリに対して個別にアクセス権を設定できる
    (他のユーザが自分のファイルを読み出したりすることを禁止するなど)
  - グループの概念を用いることで、アクセス制御とユーザ管理を分離
    ⇒ ユーザの削除なども簡単になる

- ユーザ単位でのアクセス制御が不可能なため、管理に手間がかかる
  (user1 には 許可、user2 には 拒否 といったアクセス制御は設定できない)
  → ACLにより、この問題が解消する
  　　現在のUNIXではACLがサポートされている

UNIX系のOSでは、所有者に対して、グループに対して、それら以外の第三者に対して、の3分類でアクセス権限を設定するようになっています。

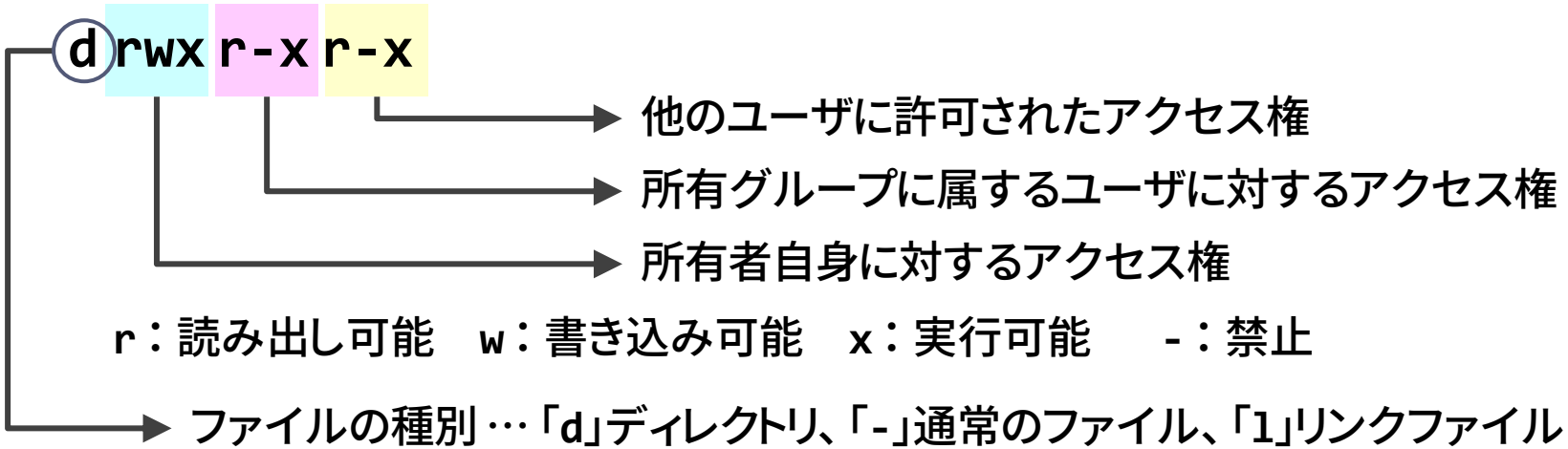
権限は、読み出し、書き込み、実行の3つです。

例：UNIXでのファイルアクセス権

- 「ls -l」コマンドでファイルに設定されているアクセス権を表示させる

```
$ ls -l
-rw-r--r--   1 user1   staff1    43   Sep   1   2008 fruits
drwxr-xr-x   2 user1   staff1  4096   Jul   3 13:57 school
```

アクセス権の設定状況　　所有者（ユーザ）　　所有グループ（ユーザが属している主グループ）



UNIXでのアクセス権限は、このように表示されます。これは、UNIXをコマンドベースのインタフェースで使うとよく見られる表示ですが、カレントディレクトリの内容をリスト表示したものです。

ここには、fruitsというファイルとschoolというディレクトリがあり、所有者はuser1で、グループはstaff1です。

fruitsは、所有者に対してrw-：読み書き可、グループのユーザに対してr--：読み出しのみ、それ以外の第三者に対してもr--：読み出しのみ、の権限となっていることが示されています。

なお、「リンク」は第9回で出てきたものです。

- アクセス権の記号の意味

記 号	意　味［ディレクトリの場合］
r	読み取り可能［ファイルの一覧情報を表示できる］
w	書き込み可能［ファイル/サブディレクトリを作成・削除できる］
x	実行可能［カレントディレクトリとして設定できる］
-	禁止（権限がない）

- アクセス権（パーミッション）の例
  - r-- ：読み出しのみ許可する
  - rw- ：読み書きを許可する
  - r-x ：実行を許可する（読み出しも許可）
  - パーミッションの設定/変更は、所有者（と管理者）だけが可能

アクセス権の情報は、iノードに記録されている

前頁で挙げたアクセス権（よくパーミッションと呼ばれます）の記号の意味はここに示す通りです。

セキュリティツールとしての暗号

- **暗号**
  平文と呼ばれる元データ（メッセージやファイル）を許可された人間だけが、平文に戻せる（復号）ような方法で暗号文に変換する
  「鍵」と呼ばれる秘密に基づいて、メッセージの送り手と受け手を限定し、通信路が信頼できなくても構わないようにする
  - 鍵によりメッセージの送り元（誰が作ったか）を検証できる
  - 秘匿したいデータを送付先（鍵をもっているもの）しか見られないようにできる、改ざんされないようにできる
  - 共通鍵暗号（秘密鍵暗号）… 暗号化と復号に同じ鍵を用いる
  - 公開鍵暗号 … 暗号化と復号に別の鍵を用いる
    理論上は、暗号化鍵を適切に選びさえすれば、それに対応する復号鍵は発見できない　→ 暗号化鍵は公開できる（公開鍵）

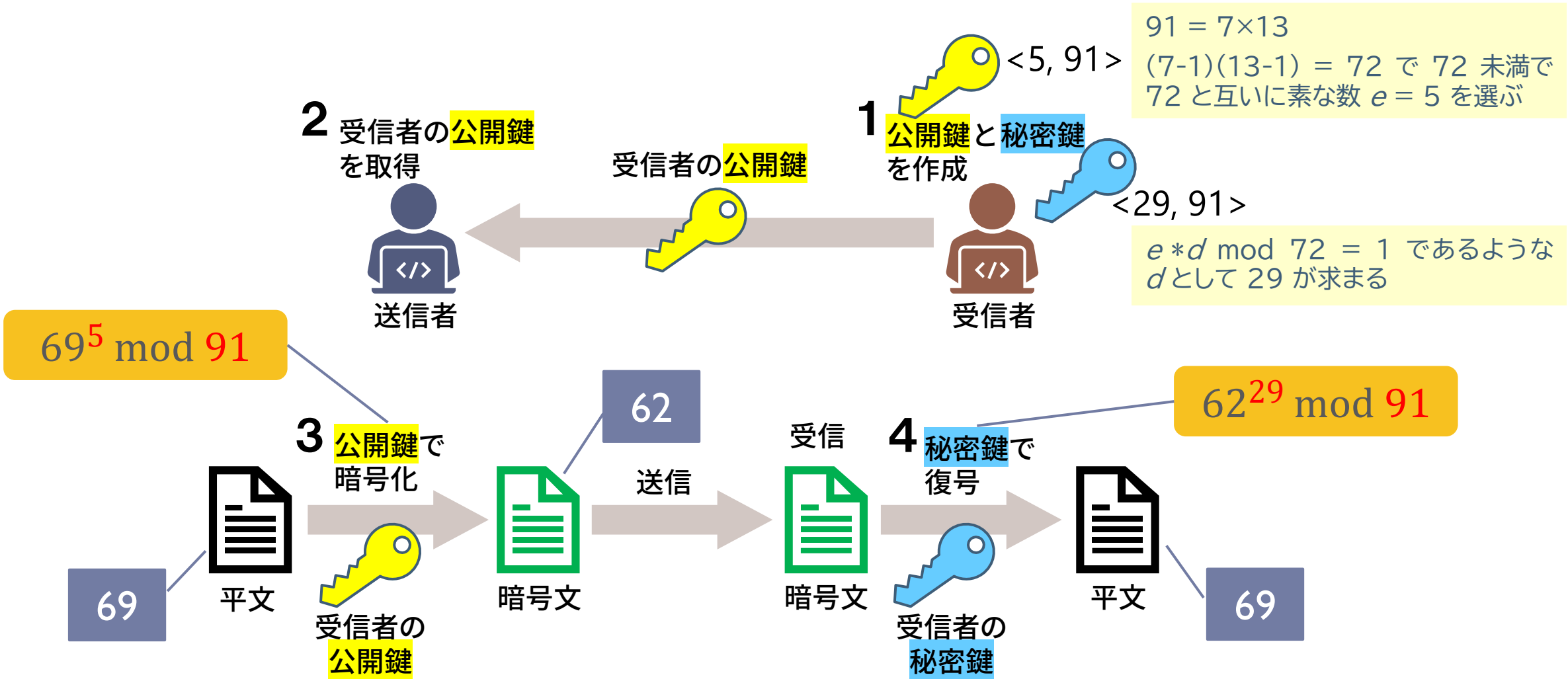
セキュリティにおいて重要な役目をもつ暗号についてです。
暗号化とは、データの内容を他人には分からなくするように変換することです。いくつかの種類があり、効率や安全性が異なります。

秘密鍵方式と公開鍵方式について説明します。

秘密鍵方式では、メッセージの暗号化と復号に要する計算量が大きくないので効率がいいですが、その反面、送り手と受け手が共有する秘密鍵を両者ともに保持しなければならないという問題があります。一方から他方へ物理的に鍵をわたさなければならないことが欠陥になります。
これに対して公開鍵方式では、理論上、暗号化鍵を適切に選びさえすれば、それに対応する復号鍵を発見できない、という特徴があり、暗号化鍵を公開することができます。（復号鍵だけを秘密にしておけばよい）
RSAと呼ばれる公開鍵の体系は、コンピュータにとって、大きな数の積を求める方が、大きな数の素因数分解よりもはるかに簡単であるという事実をもとに開発されました。
公開鍵暗号方式の一番の問題点は、共通鍵方式と比べて暗号化や復号に多大な時間がかかることです。

暗号とユーザ認証について、別途提供する資料（参考書：「オペレーティングシステムの概念」からの抜粋）を参考にしてください。

公開鍵を用いた暗号化と復号



ユーザ認証

- 計算資源へのアクセス制御を実現するため、
  - 保護したい資源にアクセス条件を付与する
  - 個々のユーザを区別し、管理する

両方の仕組みが必要

アクセス権の話（上述）

主体の保証
本人であること

- 主体はユーザとプロセス
  多くの局面でプロセスが主体となる
  - プロセスに「誰（ユーザ）が作ったか」という属性を持たせることでアクセス制御を定義することが多い
  - あるプロセスが作ったファイル、プロセスなども生成したプロセスが持っていた「誰が作ったか」という属性を持つ
  - 操作時に主体が、対象を作成した主体と一致した場合に許可し、そうでない場合は許可/不許可が必要
  - 多くの場合、ユーザやプロセスには「ID」（固有の番号）を割り振る
- ユーザの認証が最も重要になる

「誰に」の話になります。すなわち、個々のユーザを区別して本人（すなわち正しいユーザ）であることを保証する仕組みについてです。

「誰」ということですが、多くの場合、実行するのはプロセスですので、プロセスに「誰が作ったか」という属性を持たせてアクセス制御を定義することになります。
操作するときに作った人と使う対象を作った人が同じであれば許可となります。

そこで、ユーザを認証（正しいかどうか確認）することが重要になります。

- カーネルではユーザ認証を行わない
  アプリケーションの一種として実施される
  - ユーザ認証のプログラムをカーネル外で実行し、認証が終わった後で、そのオペレーティングシステムが利用可能となる
  - 認証の結果、何らかのID（ユーザID：UID）がオペレーティングシステムに通知される
- 認証の手段
  生体（指紋、静脈など）、署名、ICカード、パスワード などがある
  - パスワードが最もよく用いられている
    - パスワードは暗号化されて格納される
      （DES暗号やMD5ハッシュ関数が用いられる）
    - ログイン時、入力されたパスワードを暗号化し、登録されたものと比較

  暗号化されていても、パスワードを解読される危険性がある
  - 数字8文字なら 10^8 回の計算で見破られる

ユーザがコンピュータにログインするとき、OSは通常、そのユーザが誰であるかを決定します。この過程がユーザ認証です。実は、OSのカーネルではユーザ認証を行いません。ログインするときは、カーネルの外でユーザの認証を行って、OKであればOSが利用可能になるというわけです。

ログインしようとするユーザを認証する方法のほとんどは、「ユーザが知っている何か」、「ユーザが持っているなにか」、「ユーザ自身の何か」、に基づいて行います。
「ユーザが知っている何か」は、たとえば、パスワードがよく使われているものです。

プロトコルスタックでのネットワーク機能の保護

- 経路の暗号化、認証
  覗き見や改ざんの防止、アクセス権限の操作対象の保証
  - IPsec（Security Architecture for Internet Protocol）
    経路（IPパケット）を認証/暗号化することにより、ネットワーク層でIP通信を保護するためのプロトコル群
    暗号化がサポートされていない上位層でもセキュリティを確保できる
    - VPN（Virtual Private Network）
      – 信頼できないインターネット上で信頼できるネットワークを構築する –
      の基礎として広く用いられる
- フィルタリング
  パケットの属性に対する確認と送受信権限の規定
  - パケットフィルタリング
    送受信される特定パケットをチェックし、送受信の禁止/許可を制御（IPパケットのIPアドレス、ポート番号などに対して、パケットを受諾/破棄するなど）
    - アプリケーションで不用意にポートを公開している場合などに、OSレベルで統一的にアクセス制御できる
      … ファイアウォール

ネットワーク機能の保護の視点からは、覗き見と改ざんの防止やアクセス権限の操作対象の保証を行います。
また、パケットの属性に対する確認と送受信の権限の規定があります。

前者について、IPsecは暗号化によってネットワーク層のプロトコルを保護します。
IPsecは、OS内のプロトコルスタックないしはミドルウェア、サービスプログラム群で実装されます。

パケットフィルタリングでは、たとえば、自計算機に向けられた特定ポートのサービスを、パケットレベルで遮断し、アプリケーション層のプログラムに到達できないようにできます。

教科書との対応など

- アクセス権について、アクセス制御行列やアクセス制御リストの話を追加しています
- ネットワークの保護について追加しています
- ファイルの保護については、内容の破壊に対処するための手法もあります
  - 多重化ファイル（ミラーリング、デュプレックス）
  - バックアップ
- 13.3 記憶保護と実行モード については、仮想記憶やハードウェア支援の回で説明しています
- 認証について、スライド資料ではユーザの認証を説明していますが、認証は対象の正当性や真正性を確かめることで、
  - データが改ざんされていないことの確認（データ認証）
    ハッシュ関数による
  - 送り手/本人の確認
    デジタル署名による（公開鍵暗号が用いられます）

  もあります

教科書での説明と記載箇所が違うところがありますので、読むときに注意してください。

第11回の課題

(1) AさんがBさんの公開鍵で暗号化した電子メールを、BさんとCさんに送信した結果のうち、適切なものはどれか

ここで、Aさん、Bさん、Cさんのそれぞれの公開鍵は3人全員がもち、それぞれの秘密鍵は本人だけがもっているものとする

(平成30年度 春期 基本情報技術者試験)

ア　暗号化された電子メールを、Bさんだけが、Aさんの公開鍵で復号できる
イ　暗号化された電子メールを、Bさんだけが、自身の秘密鍵で復号できる
ウ　暗号化された電子メールを、Bさんも、Cさんも、Bさんの公開鍵で復号できる
エ　暗号化された電子メールを、Bさんも、Cさんも、自身の秘密鍵で復号できる

(2) pp.11-12 に示すアクセス制御方式をもつファイルシステムにおいて、ファイルに対して以下のアクセス権の条件すべてを満足する設定(アクセス権の設定状況)を示せ

- 全ての利用者が実行できる
- 所有者、および所有グループの利用者だけが読み出しできる
- 所有者だけが書き込みできる

今回の課題です。クラスウェブのレポートで提出してください。

期限は、6/29 の午前中とします。

事後学習・事前学習

- 今回の講義資料に基づいて内容を振り返り、教科書などの該当箇所を読む
- 教科書第2章に目を通しておく

今回の講義内容の振り返りと次回の準備をお願いします。