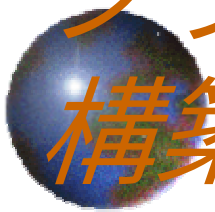
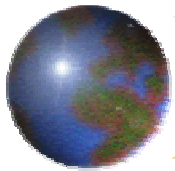
インターネットにおける真にプライベートなネットワークの構築



東京大学　　光来健一
筑波大学　　千葉滋

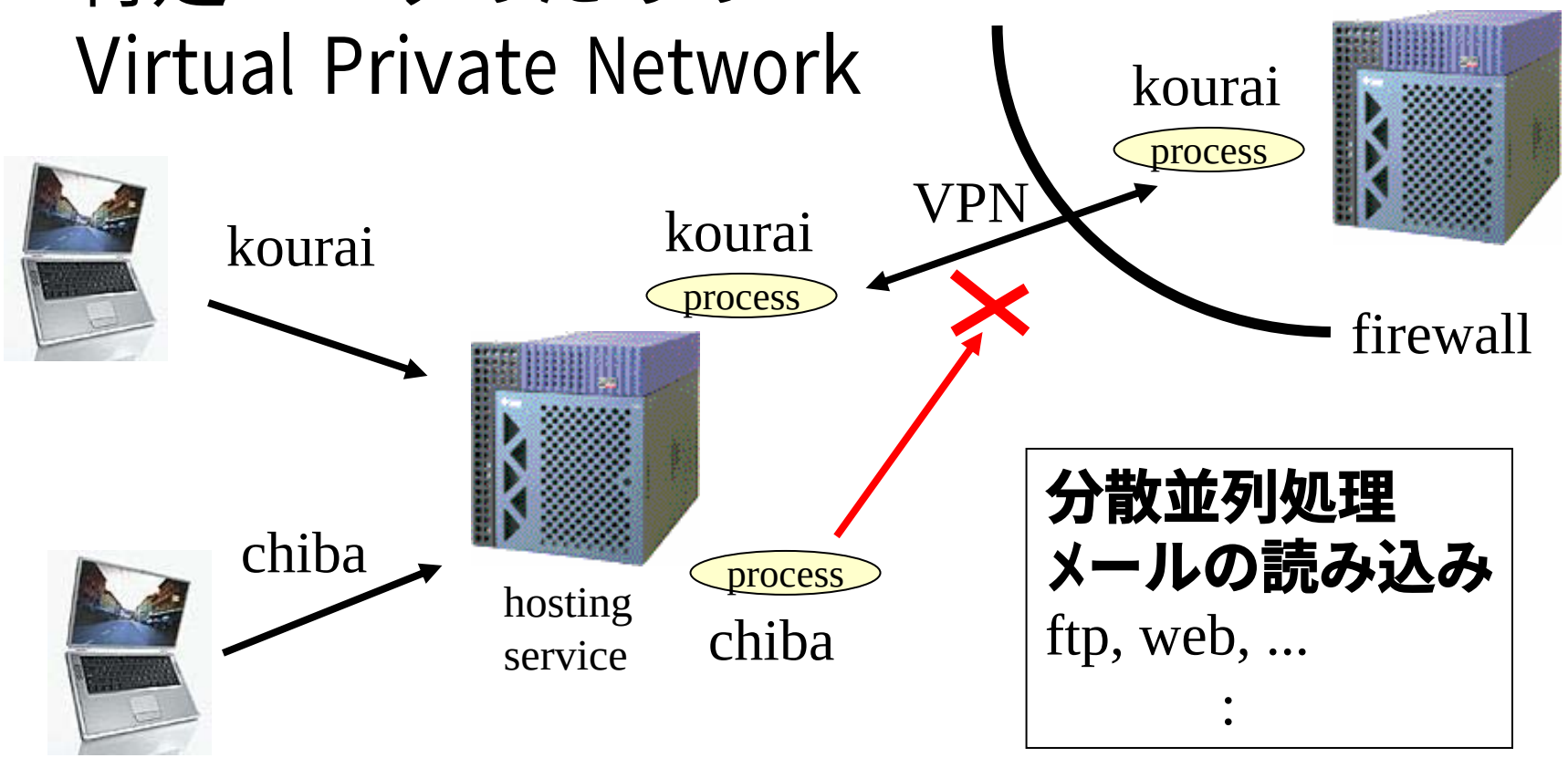


パーソナルネットワーク

- 特定のユーザだけが使えるネットワーク
- OSが提供する機能
  - あるネットワーク資源について、
    特定ユーザのプロセスだけにアクセスを許可

パーソナルVPN

- 特定ユーザのための
  Virtual Private Network



分散並列処理
メールの読み込み
ftp, web, ...
:

既存のVPNの問題点

- 同一ホストからであれば、全てのユーザが使えてしまう
  - IPsec
    - ホスト間の認証のみ
  - PPTPやプロキシによるport forwarding
    - 認証に成功したユーザだけがVPNを作れるが、作られたVPNは誰でも使える

セキュリティ上の問題

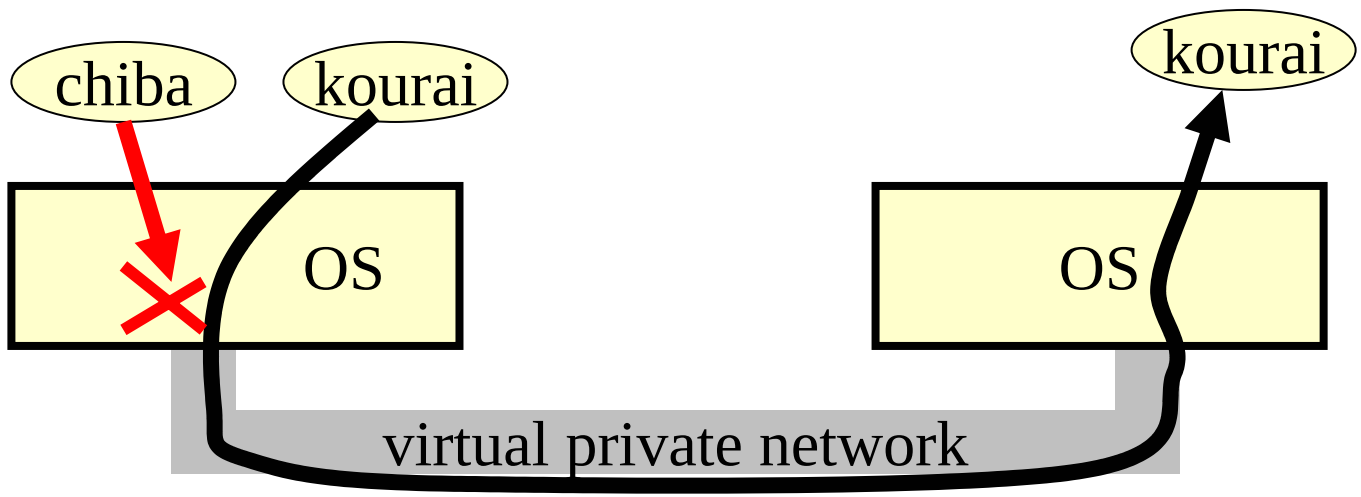
- IPアドレスによるアクセス制限
  - ウェブの内部ユーザ向けのページ
    - 130.158.85.* だけアクセスを許可
  - TCP Wrapper
    - 130.158.85.* ならpopへのアクセス許可
  - VPNによってセキュリティ・ホールができる
    - 例えば、port forwardingをするとリモートホストからのアクセスでも内部からのアクセスに見える

現実的な対処

- アプリケーション毎にVPNを作成
  - SSL (Secure Socket Layer)
  - 問題点
    - 個々のアプリケーションが対応しなければならない
    - アプリケーション毎にセッションを確立しなければならず、資源を浪費しがち
      - 最初の認証は重い

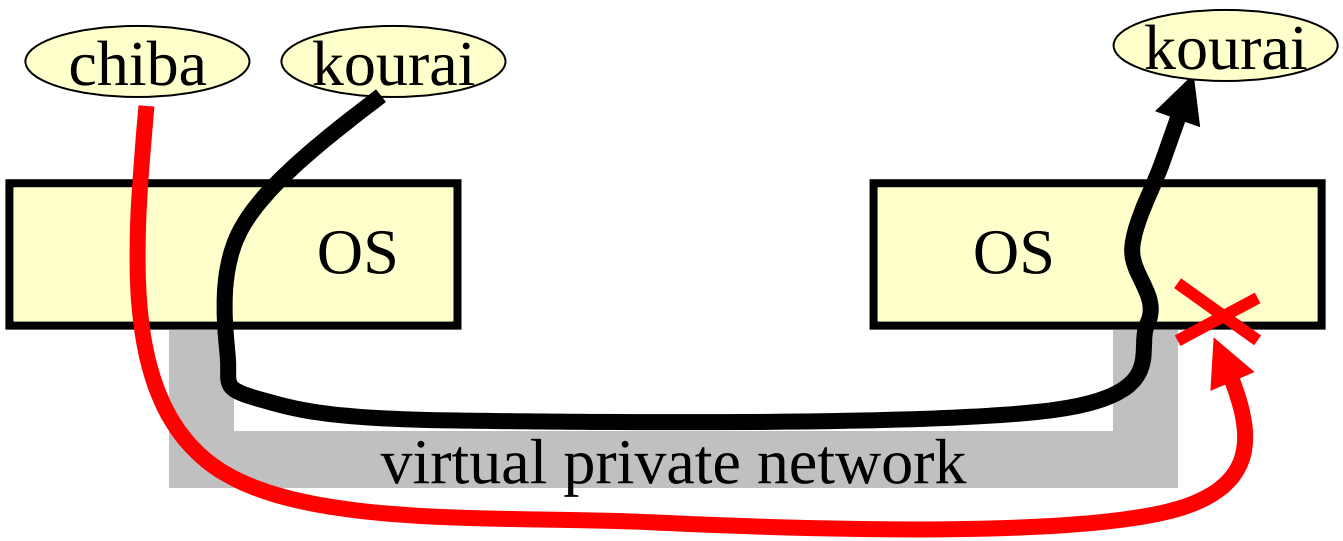
OSによるアクセス制限(1)

- プロセスのユーザを考慮
  - VPNの入り口で制限
    - 双方のOSが信頼できる場合
    - ユーザIDによりVPNの利用を制限



OSによるアクセス制限(2)

- VPNの出口で制限
  - 送り元のOSが信頼できない場合
    - 登録されていないホスト、DHCP
  - リモートユーザ認証によりVPNの利用を制限



リモートユーザ認証の実装

- TCPオプションを用いて、TCPコネクションの3-way handshakeと同時に認証を行う
  - なるべく早い段階で許可されない通信をブロックする
  - ユーザ認証に必要なパケット数をなるべく減らす

ユーザの公開鍵の送信

- クライアントはSYNパケットでユーザの公開鍵のハッシュ値を送る
  - ハッシュ値はIPアドレス、ポート、シーケンス番号も含めて計算し、攻撃されにくくする
  - サーバは存在しないユーザや接続が許可されていないユーザなら、接続を拒否できる

チャレンジの送信

- サーバはSYN+ACKパケットでチャレンジを送る
  - 受け取ったハッシュ値に対応する公開鍵でセッション鍵(乱数)を暗号化する
  - TCPオプションには入りきらないので、データ部に入れる

レスポンスの送信

- クライアントはレスポンスをACKパケットで送る
  - 受け取ったチャレンジを秘密鍵で復号し、そのハッシュ値を計算してレスポンスを生成する
  - レスポンスが正しければ、ユーザ認証が成功する
    - 正しくなければ、接続を切断する

リモートユーザ認証の簡略化

- 一旦ユーザ認証した後は簡略化できる
  - ユーザの公開鍵を送る際に、以前に交換したセッション鍵のハッシュ値も送る
    - ハッシュ値はシーケンス番号も含めて計算する
      - リプレイ攻撃を防ぐ
    - セッション鍵が正しければ認証完了
      - 期限切れなら通常の認証を行う

通信の暗号化の実装

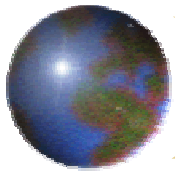
- 準備段階としてSSL Wrapperを実装した
  - カーネルでアプリケーション透明にSSLを実装できるかどうか検証
    - どの層でどの暗号を用いるのが良いかは現在、模索中
  - OpenSSLライブラリを用いた

SSL Wrapper

- システムコール関数をフックし、透過的にSSLを用いて暗号化通信を行う
  - LD_PRELOADで指定したライブラリを先にロードし、libcの関数をオーバロードする
    - connect → connect + SSLconnect
    - accept → accept + SSLaccept
    - read, write → SSLread, SSLwrite
    - …

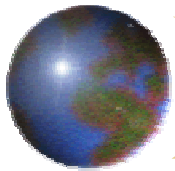
SSL Wrapperからの知見

- SSLacceptはコネクションが張られた時点で行う
  - acceptを発行した時点ではデッドロックする場合がある(ftp)
- SSLacceptが完了してからlistenソケットをselectで返す
  - acceptするとブロックする場合がある
- non-blockingソケットでのconnectへの対応
  - connectはしたが、SSLのネゴシエーションが完了していない場合がある



関連研究

- ssh X11 forwarding
  - Xの認証機構を使ってポートの使用を制限
- IPsec
  - Security Association(SA)をユーザ毎に割り当てることも可能かもしれない
- Ident
  - TCPコネクションを張ったユーザを問い合わせるプロトコル



まとめと今後の課題

- パーソナルVPNを提案した
  - OSによるユーザのアクセス制限を行う
- 今後はカーネルで暗号化を行えるようにしていく