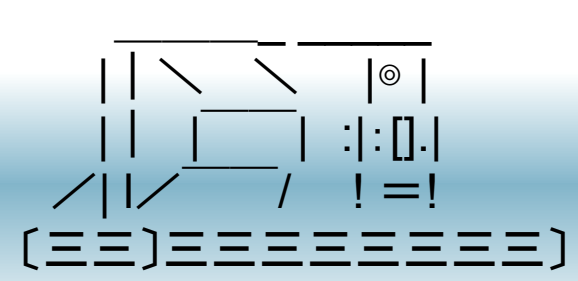
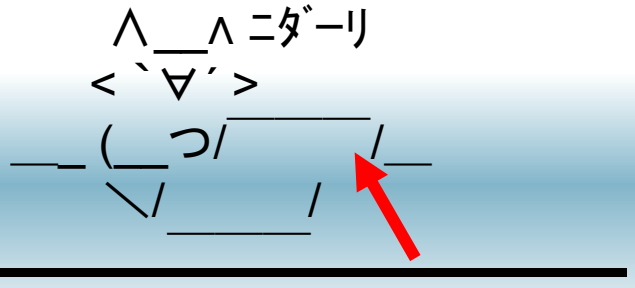
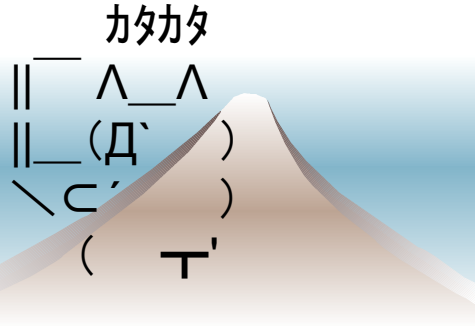
認証技術の必要性

- 何も考えずに認証すると・・・・
  - パスワード
    - 普通に盗聴されて盗まれる
  - IDカード
    - IDカードが送る情報を盗まれて真似される
  - 指紋
    - 指紋のデータ情報を盗まれて（ry







認証技術

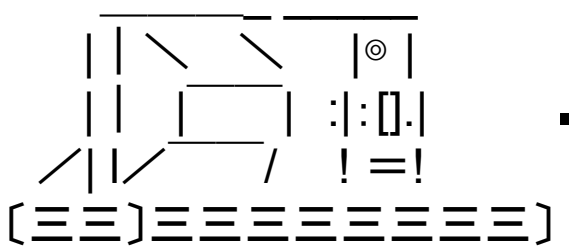
- Challenge and Response
- One Time Password
- Message Authentication Code（MAC）
- デジタル署名

- 上記は同列のものではない、念のため・・・

盗聴されても大丈夫なためには？

- Challenge and Response
  - サーバ側が乱数で生成した何らかのデータを送る
  - 送られてきたデータを処理して送り返す
  - サーバ側は送り返されたデータを検証

Challene and Response

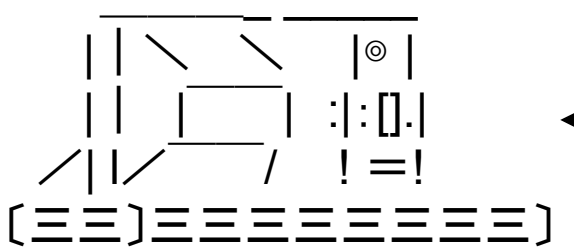


「ABC」という文字列

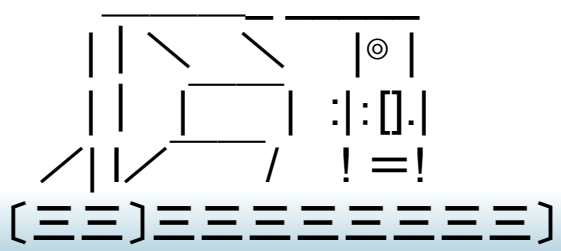
文字列を共有鍵（or秘密鍵）で
暗号処理

「ABC」→「CDEFG」

Challenge and Response



「CDEFG」という文字列

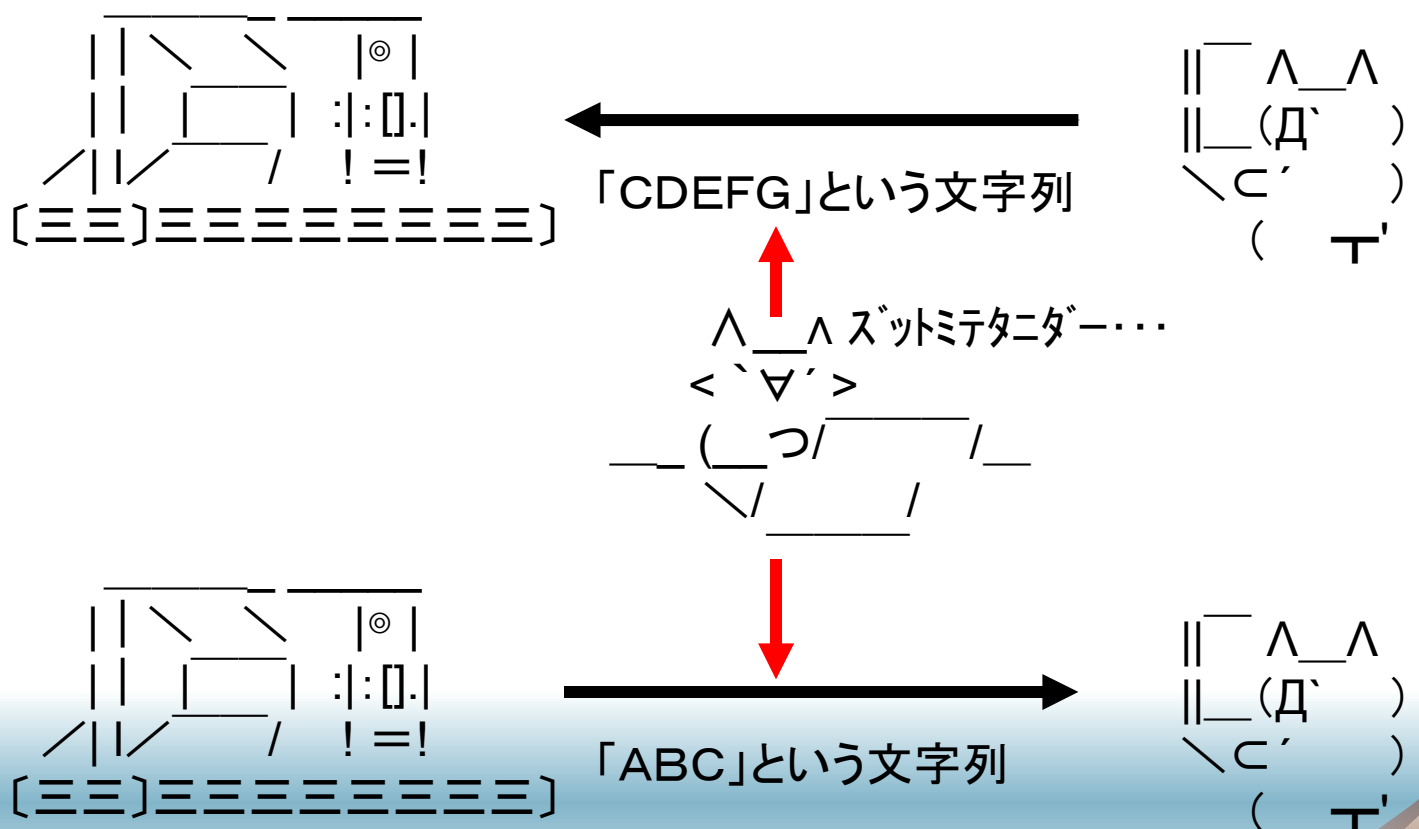


文字列を共有鍵 or 公開鍵で検証

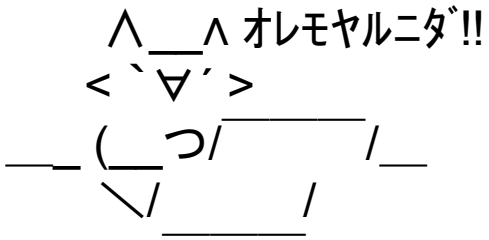
「ABC」→「CDEFG」

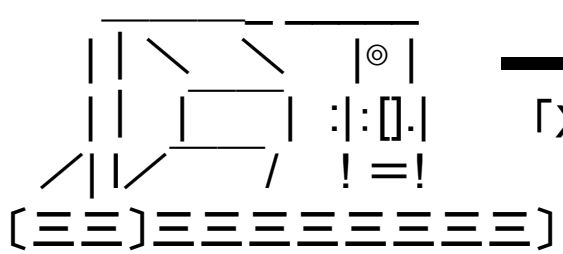
認証ok！

Challenge and Response

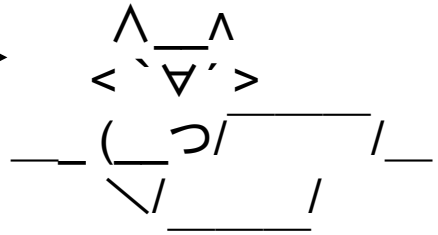


Challenge and Response

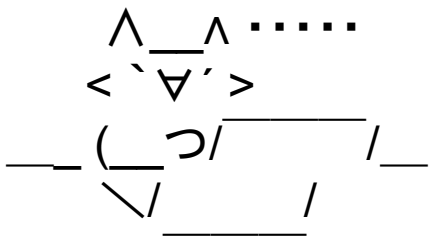


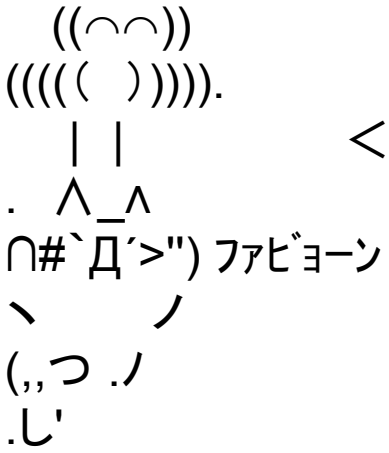


「XYZ」という文字列



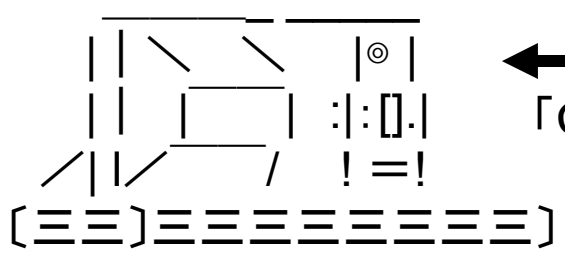
Challenge and Response



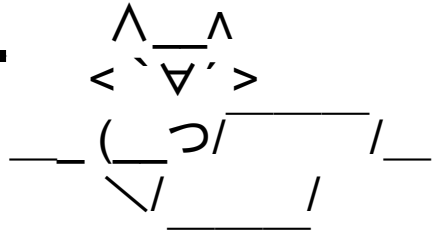


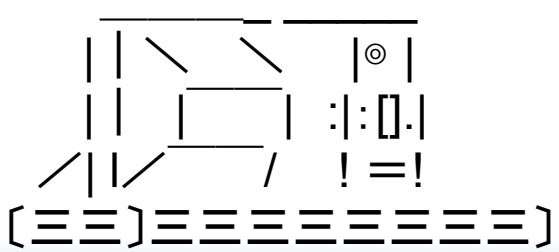
＜何送ればいいかわからないﾆﾀﾞ！！

Challenge and Response



「CDEFG」という文字列





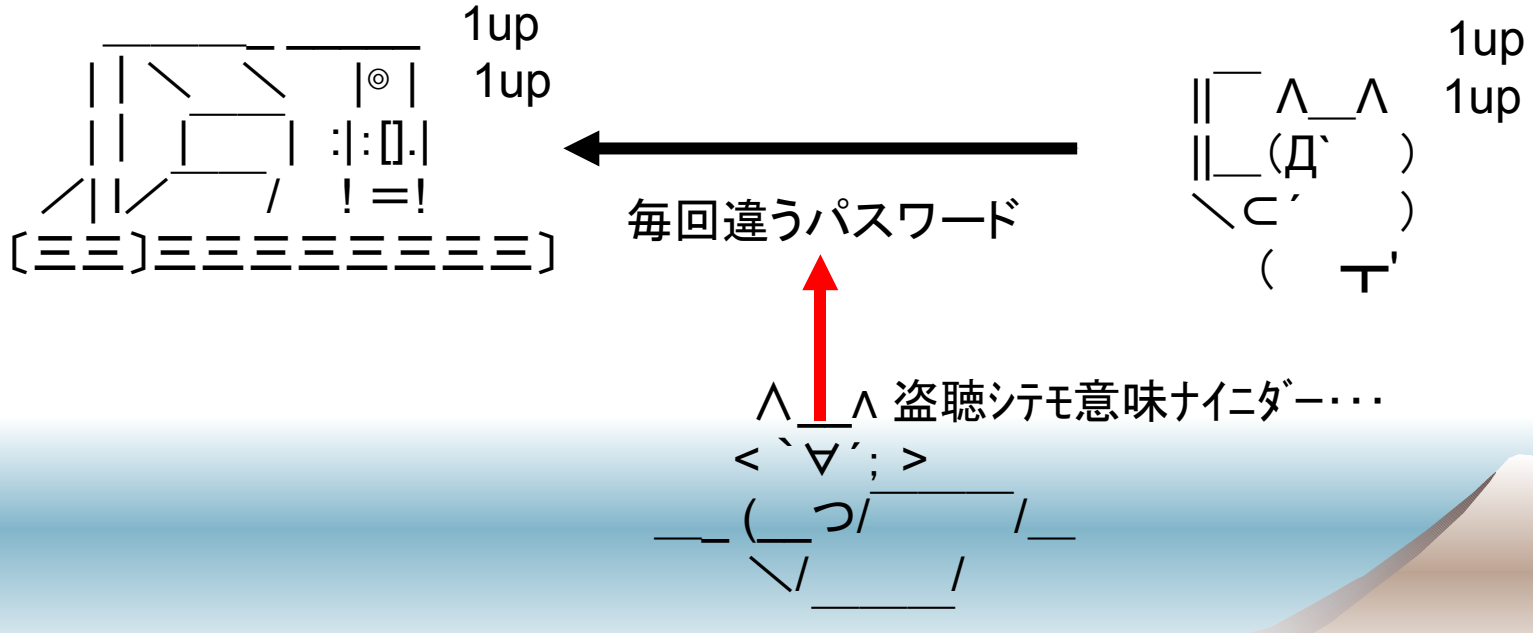
文字列を検証

「XYZ」→「VZDEG」

認証失敗！

One Time Password

- パスワードを使い捨てにする
- 一回使うたびに（暗号化処理などで）次のパスワードを双方で計算

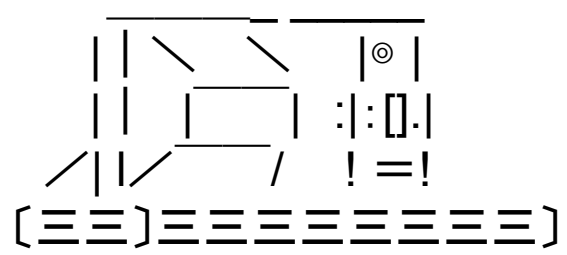


暗号技術のおさらい

- 共有鍵暗号
  - 一つの鍵で暗号化・復号化する方式
- 公開鍵暗号
  - 公開鍵と秘密鍵で暗号化・復号化する方式

Message Authentication Codeとデジタル署名

- 検証するときに・・・・



送った文字列：「ABC」
戻った文字列：「CDEFG」

＊MAC
共有秘密鍵で送った文字列を暗号化し、
戻ってきた文字列と比べる

「ABC」→「CDEFG」

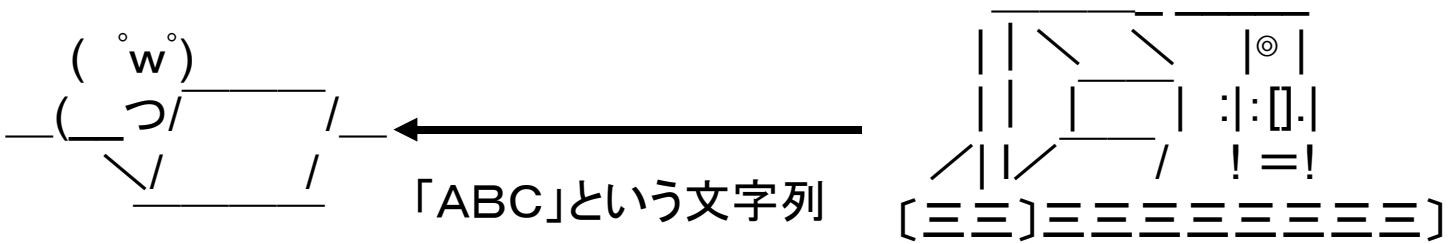
＊デジタル署名
公開鍵で戻ってきた文字列を復号化し、
戻ってきた文字列と比べる

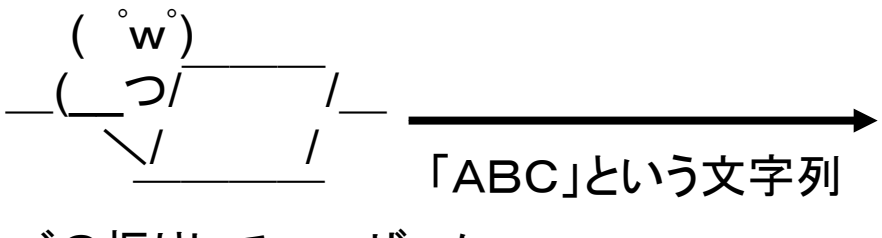
「CDEFG」→「ABC」

しかし・・・・

- 実は Challenge and Response も万能ではない
- 中間者一致攻撃

中間者一致攻撃



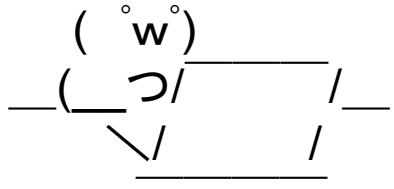


サーバの振りしてユーザーに
送信

中間者一致攻撃

文字列を共有鍵(or秘密鍵)で
暗号処理

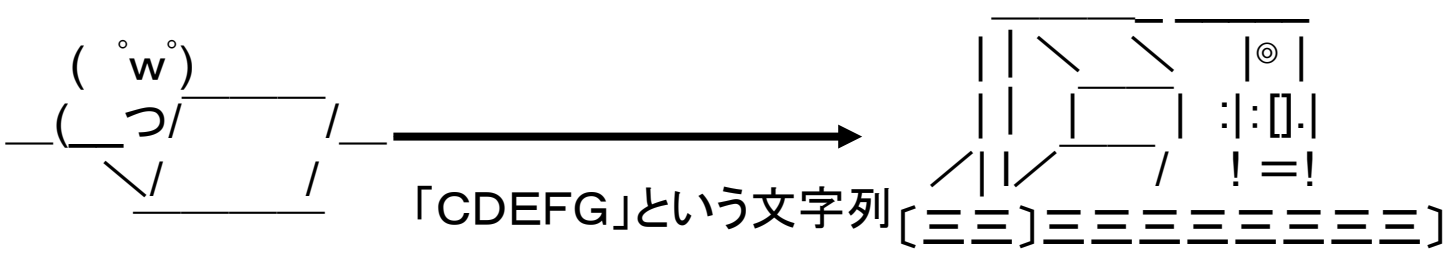
「ABC」→「CDEFG」



「CDEFG」という文字列

サーバの振りしてユーザー
から送られてきたデータを受信

中間者一致攻撃



あらためてサーバに
データを送る

攻撃成功！

中間者一致攻撃

- Challenge and Response でも、OTP でも、サーバー側を詐称することで簡単にパスワードを盗めてしまう
- つまり、クライアントがサーバ側を認証する必要もある

おまけ：APOP脆弱性の話

- 今日のニュースで「APOPでの脆弱性」
  - あれもサーバ詐称でパスワードが盗めますよ、という話
  - 別に新しい話ではないのに何で今頃報道されるか謎
  - しかも某紙では「暗号化のカギとなる「MD5」を逆にさかのぼり、暗号化する前のパスワードに戻す手法を見つけた」 → ハッシュの逆演算？

(゜Д゜)ﾊｧ?

おまけのおまけ
弱衝突と強衝突

- 弱衝突
  - あるハッシュ値と同じ値を持つハッシュ値を見つける
- 強衝突
  - 同じハッシュ値を持つ二つ以上の組を見つける
- Birthday Paradox
  - 自分と同じ誕生日を持つ人が30人のクラスに居る確率は約 1/12
  - 30人クラスに同じ誕生日の人が居る確率は50%

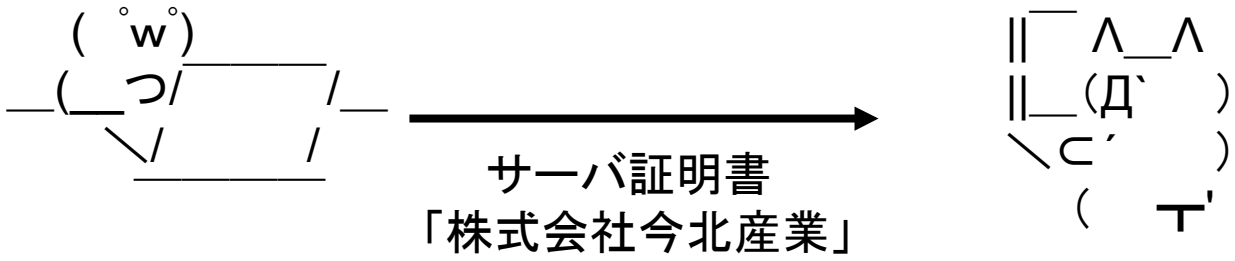
サーバ認証

- クライアントはどうやってサーバを認証するか？
  - サーバ証明書（公開鍵暗号）でメッセージを署名してもらう
- でも、サーバ証明書の検証はどうやるの？

認証局（CA）

- いろいろな企業・個人に証明書を発行する機関
- クライアントは認証局の証明書を使ってサーバ証明書を検証する
- 自分のブラウザのCAを見てみましょう
  - IE→「インターネットオプション」→「コンテンツ」→「証明書」→「信頼されたルート証明機関」

サーバ証明書



PCに入ってるCAの証明書で
サーバ証明書を検証

偽者発覚！！

証明書のしくみ

- 基本は公開鍵暗号
  - 認証局も、各証明書を発行された組織も自分の秘密鍵・公開鍵を持っている
  - 組織は公開鍵を認証局に送る
  - 認証局は公開鍵を署名して送り返す
  - 第三者は認証局の公開鍵を利用してその組織の公開鍵を検証できる

証明書のしくみ

認証局

証明書

認証局の（秘密鍵による）
署名がついた証明書

認証局の（秘密鍵による）
署名がついた証明書

認証局の公開鍵
（最初からブラウザに
入っている）

公開鍵で検証できる

証明書を検証してみよう

- https://www.amazon.co.jp/ を見る
- 右下の鍵マークをクリック
  - 詳細キーなどで証明書を見てみよう

まとめ

- ネットワークの盗聴
  - 簡単に盗聴できる
- 認証
  - さまざまな認証技術
  - 盗聴・なりすましされても安全な方法
  - 認証局