インターネット通信の安全性検査全自動化へのアプローチ

産業技術総合研究所
大岩 寛

NICT オープンハウス 2013

インターネット通信の安全性検査全自動化へのアプローチ

何のお話？

- インターネットで安全に通信したい
  - インターネットには悪者もいます
  - 盗聴や改ざんを防ぎたい
- 安全な通信のためのソフトを使おう
  - みんなで安全な通信の方法を考えました
- ソフトが間違っていると安全にならない
- ソフトが正しいことを確かめたい
- 確かめる方法を作りました
  - 特に、自動化と完璧さにこだわりました

インターネットと安全な通信

インターネットとセキュリティ

- 日常生活の一部としてのインターネット
  - 電子メール
  - 電子商取引
  - 電子政府
  - ソーシャル
    ネットワーキング

  - 個人情報や
    経済情報が
    ネットワーク上で
    流通する時代

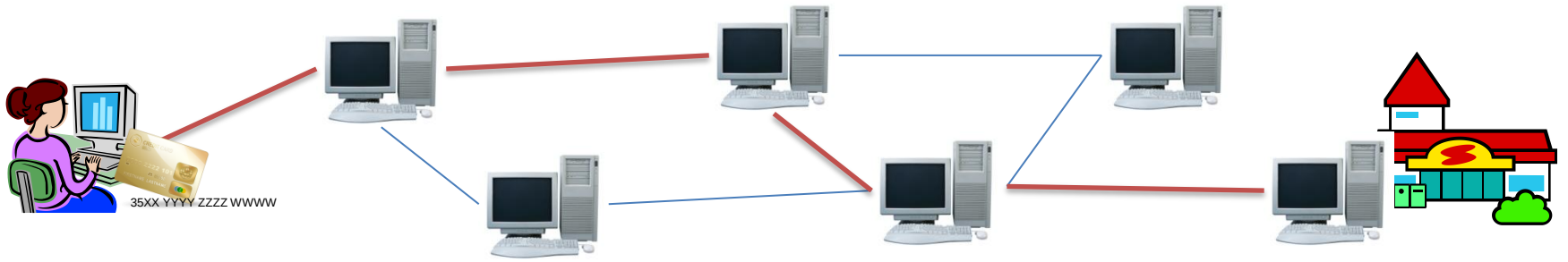
インターネットとセキュリティ

- インターネット利用上の我々へのセキュリティ脅威
  - コンピュータへの不正ソフトの侵入
    - コンピュータウィルス、マルウェア
  - ネットワーク上の通信の盗聴や改ざん
  - 偽サイト（フィッシング）

今回のトピック

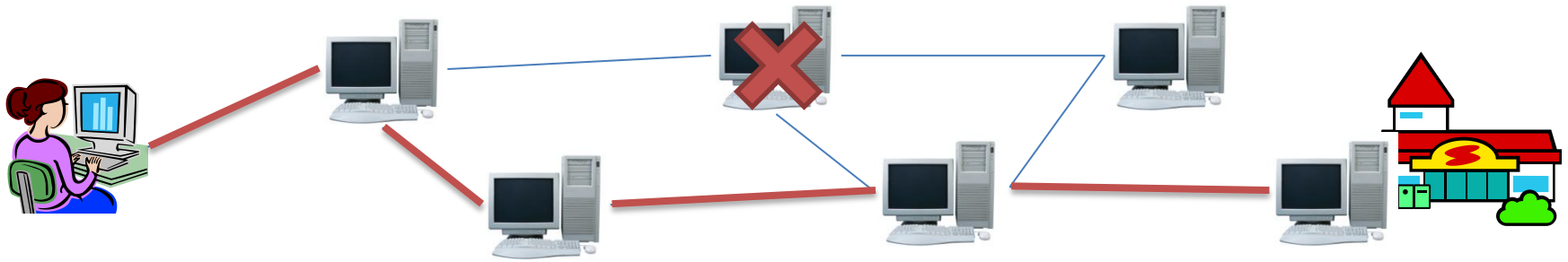
インターネットとセキュリティ

- インターネット通信の特徴
  - ネットワーク上を「リレー」して転送される通信データ
    - 利点
      - コストが安い（自分で長い通信線を張る必要がない）



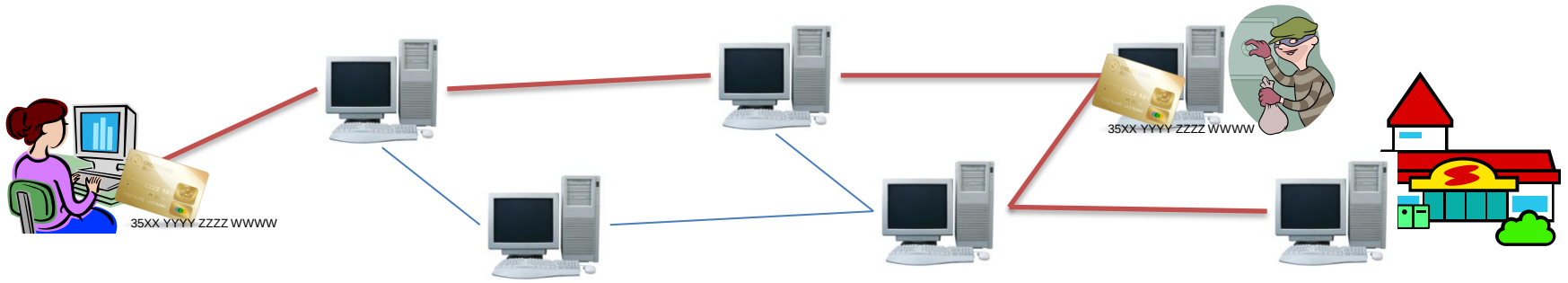
インターネットとセキュリティ

- インターネット通信の特徴
  - ネットワーク上を「リレー」して転送される通信データ
    - 利点
      - コストが安い（自分で長い通信線を張る必要がない）
      - 障害に強い（複数の経路が確保できる）



インターネットとセキュリティ

- インターネット通信の特徴
  - ネットワーク上を「リレー」して転送される通信データ
    - 利点
      - コストが安い（自分で長い通信線を張る必要がない）
      - 障害に強い（複数の経路が確保できる）
    - 欠点
      - 通信途中で他人に妨害・覗き見される可能性がある
        - » 途中誰の計算機を経由するかは制御できない

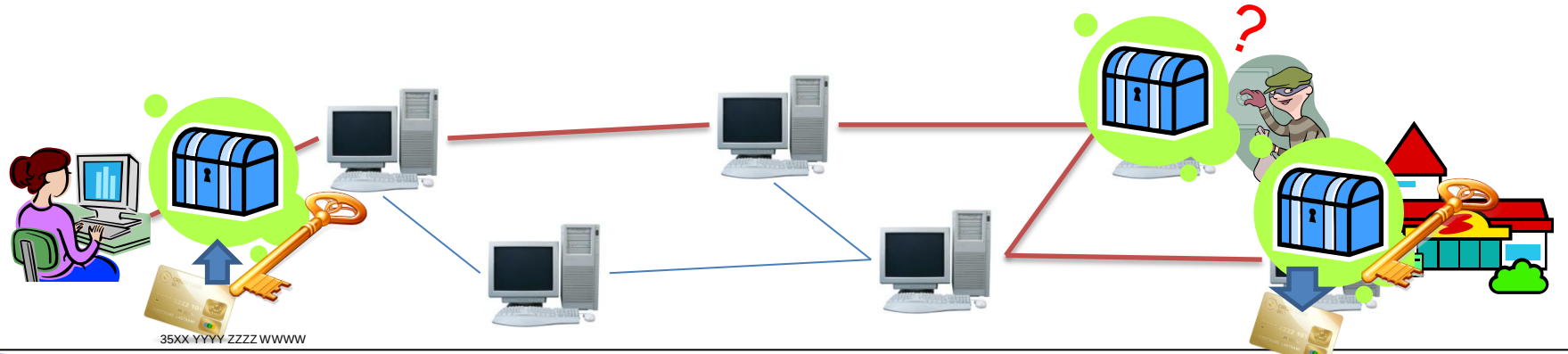


インターネットとセキュリティ

- 悪者に覗き見されると困る情報
  - 経済的な情報
    - 銀行口座番号
    - クレジットカード番号
    - 会員番号・ユーザID・パスワード　　など
  - 個人情報
    - 氏名・年齢・性別・住所
    - 個人の趣味・嗜好
    - 他人への私信　　など
  - これらを安全にやりとりするにはどうする？

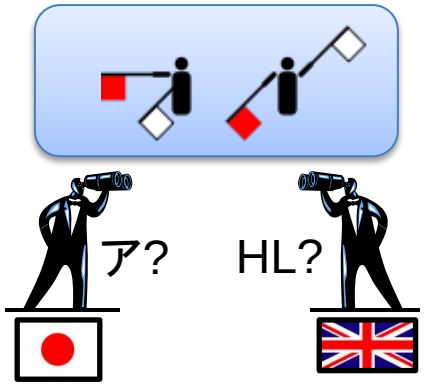
答え: 暗号化

- 暗号技術の利用
  - 通信の当事者（普通は2名）のみが
    内容を理解でき、他人には
    通信があることしかわからないようにする
    - 「鍵」による「錠付き箱」のようなものと
      大まかには考えればよい



暗号通信と規約

- お互いが「同じやり方」で
  暗号化の処理をしないと通信ができない
  - 暗号の種類
  - 最初に「鍵」をお互いに持つやり方
  - 通信相手の確認の仕方
  - 合意形成の手段が必要



- 解決:「標準規約」による暗号通信
  - みんなで合意したやり方で暗号通信をする
    - 1つのソフトウェアでいろいろな通信ができる
    - 安全性をいちいち確かめなくて良い

暗号通信の規約

- TLS (SSL)
  - Webブラウザにおける暗号通信の標準
    - メールなどにも利用
  - IETF (Internet Engineering Task Force) で標準化
    - 旧規格である SSL 3.0 までは米 Netscape 社が策定
    - TLS 1.0 として改訂・インターネット標準化
    - 現在の最新版は TLS 1.2 (RFC 5246)
  - 主要なソフトウェアにすべて搭載
    - Webブラウザ: Internet Explorer, Apple Safari, Mozilla Firefox, Google Chrome, Android, ……
    - Web サーバ: Apache, Microsoft IIS, nginx, ……
      - 別の会社のソフト同士でも基本的に問題なく通信できる

暗号通信の規約

- （余談）IETF?
  - インターネットの標準を決める団体
    - 法律的にはアメリカの非営利団体 ISOC の下部組織
  - オープンな標準策定
    - 誰でも自由に議論に参加できる
      - 1国1票のISOや、1企業1票の団体とは違う形態
  - 年3回のミーティング
    - 毎回1200人程度が世界中から参加
    - 前回は今月頭、バンクーバーで開催
      - 次回は3月、ロンドンで開催
      - 2015年には横浜で開催予定（日本で3回目）

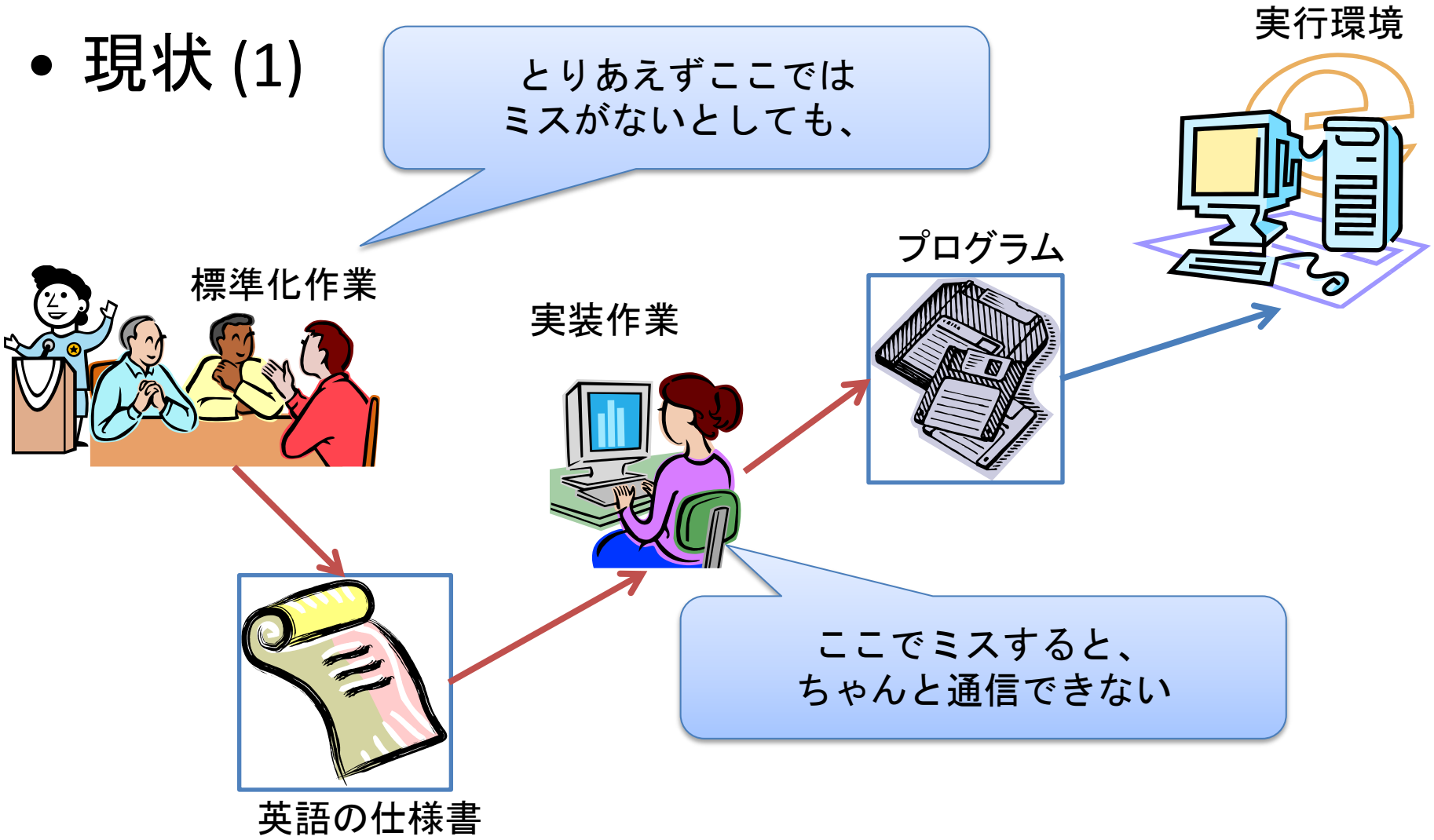
暗号通信の規約

- TLS (SSL)
  - 問題はないのか?
    - もちろん性能とか、細かい問題もあるけど…
  - 膨大な仕様書
    - TLS 1.2 (RFC 5246) は 104 ページもある
      - Web 通信の基本規約 HTTP 1.1 は 178 ページもある
    - 基本的には全部が英語（自然言語）
      - 人間が読んで、人間がソフトウェアを作るのが前提
  - *もし作ったソフトに間違いがあると…*
    - うまく通信できない（不意なエラー発生）
    - 通信の秘密が破れる…かもしれない

通信ソフトの正しさの検査

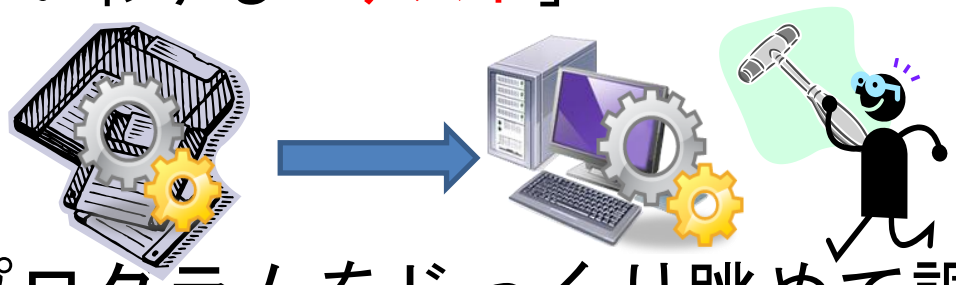
通信ソフトの作られる過程

- 現状 (1)



ソフトの検査の方法

- すごくおおざっぱには、2通りの方法
  - 実際にいろいろ走らせて試してみる
    - いわゆる「テスト」



  - プログラムをじっくり眺めて調べる
    - いわゆる「検証」

ソフトの検査の方法

- すごーーーくおおざっぱな喩え
  - 例: 入力を2倍するプログラム
  - テストの考え方
    - 1 を入れて、2が出てくることを確かめる
    - 2 を入れて、4が出てくることを確かめる
      - 完璧ではないが、複雑な物でもある程度扱える
  - 検証の考え方
    - プログラムの文面 x → (x + x) を見る
    - 入力を x としたとき、

      出力が x の2倍であることを数学的に証明する
      - 完璧を目指せるが、問題が複雑だと大変

ソフトの検査の方法

- 検査結果の信頼性・利便性
  - テストの場合
    - 間違いが見つかった場合: 直しやすい
    - 間違いが見つからなかった場合: 信頼性に悩む
      - テストが足りないだけかもしれない?
      - 「3を入れたらちゃんと6が出てくるかなぁ？」
  - 検証の場合
    - 間違いがある場合: 見つけるのが面倒
      - 「いつまでたっても証明ができない」
      - 「ひょっとしてどこか間違ってるかも？」
    - 間違いがなかった場合: 信頼性は高い

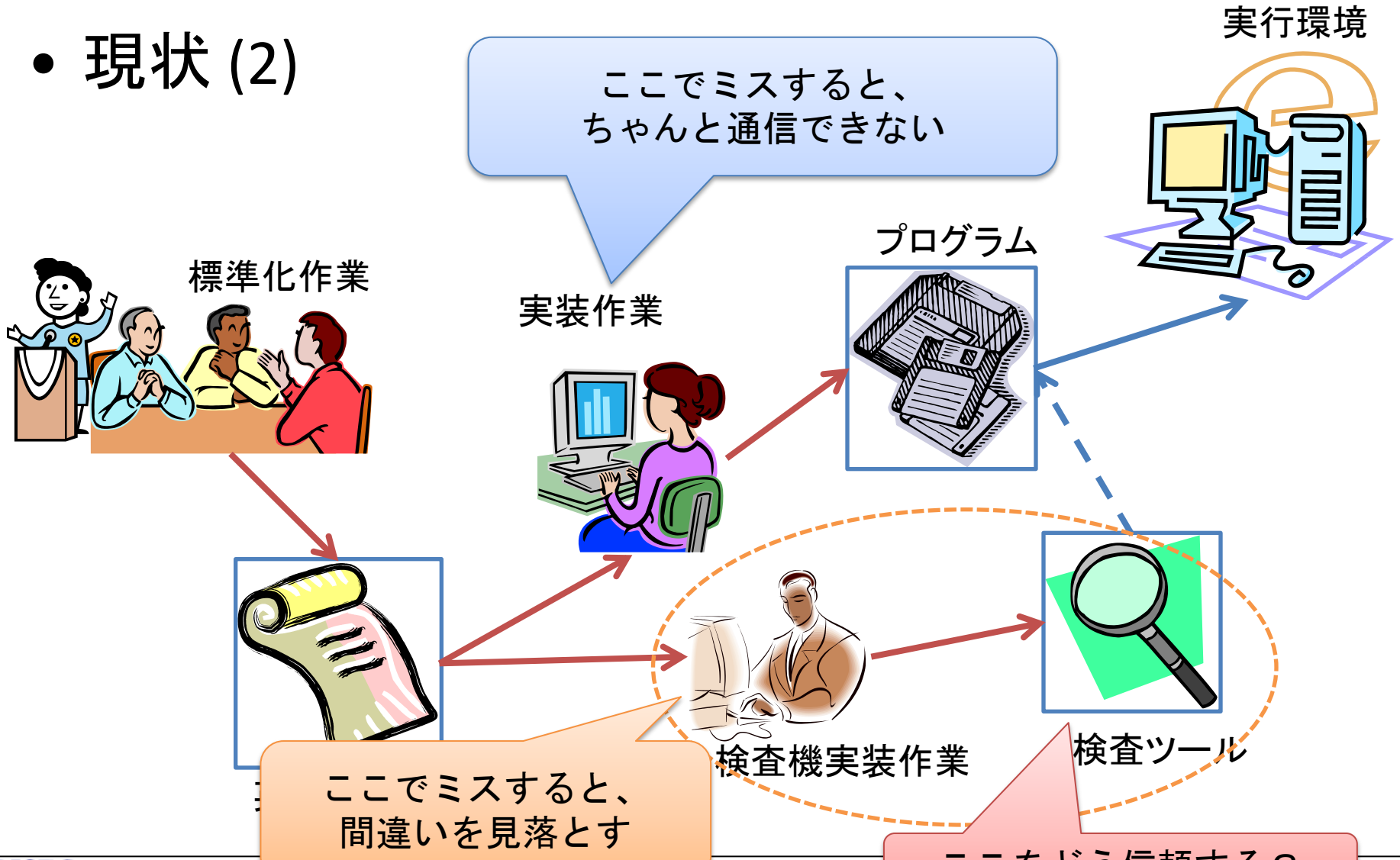
通信ソフトの検査（テスト）

- 通信するソフトウェアの場合
  - 実際に通信させて確かめる（テスト）
    - 通信相手をソフトウェアとして作る
    - 実際に通信させてみる
    - 通信の経過を監視して動作を確認する

- 暗号通信ソフトの場合の問題
  - 通信を眺めてもわからないことが多い
    - なんせ「暗号化されている」
  - 動作が複雑なので、「どれだけテストしたか」がわかりづらい
    - そもそも「テストが正しいか」も信頼しづらい

ソフトの検査の方法

- 現状 (2)



今回やってみたこと

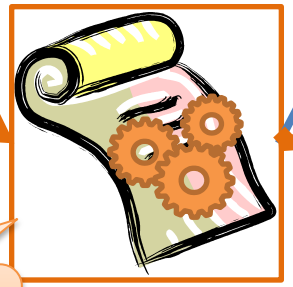
- 検査ツールを自動生成してみた
  - 検査ツールを作る段階でのミスを排除
  - 人間では作るのが面倒な数のテストを自動生成
    - 今回の範囲だけで2000通り以上
  - このために、仕様書を改良した
- 検査も自動化してみた
  - 2000通り強を30分位で検査できる仕組み
  - 暗号化処理で通信内容が毎回変わっても、ちゃんと検査を完了できる仕組み

解決しようとする問題

- 今回やったこと

標準化作業

英語の仕様書

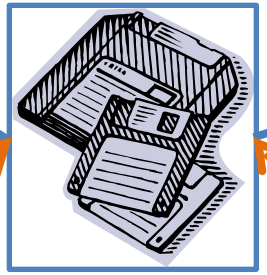


機械可読の仕様書

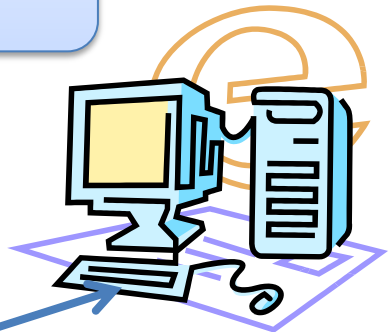
自動処理できる
仕様書を作った

実装作業

ここでミスすると、
ちゃんと通信できない



プログラム



実行環境

人手よりずっと多く
テストが網羅できる

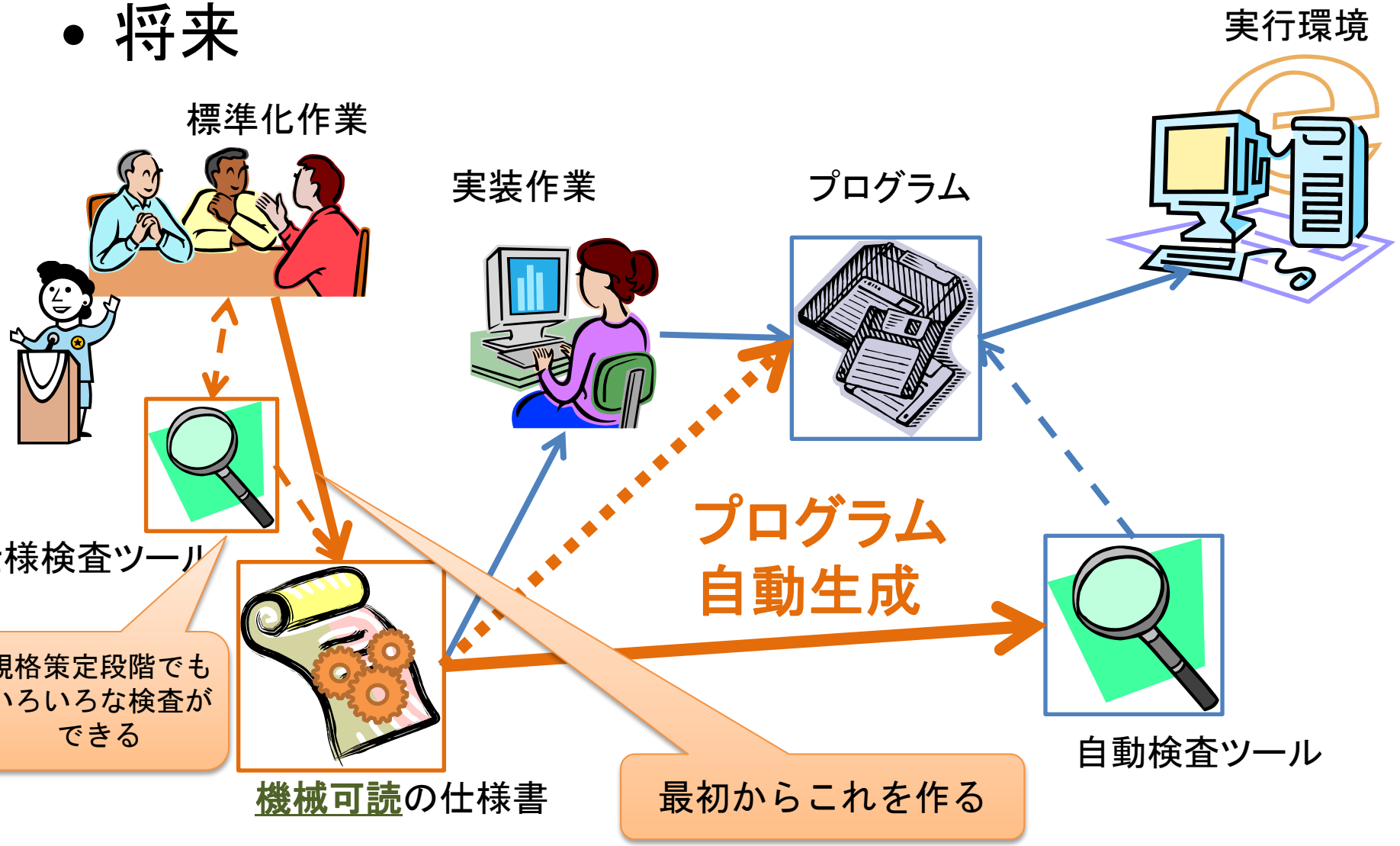
検査ツール

自動生成

ここでミスがないので、
信頼性が上がる

解決しようとする問題

- 将来



今回やったこと

- 具体的な取り組み (1)
  - 機械可読な仕様記述言語を新たに開発
    - 人間も読める・書ける
    - コンピュータでも自動処理できる
  - ここから、プロトコル検査器を自動生成
    - 抜け・漏れのない網羅検査器を実現
      - 「テスト成功時の信頼性」に一定の基準を担保
  - 「プログラムの巻き戻し」機能を使った検査
    - 2000通り位の動作を巻き戻ししながら順番に全部網羅検査するシステムを開発

今回やったこと

- 具体的な取り組み (2)
  - この仕様記述言語でTLSの一部を実際に記述
  - 世の中の4つ以上のプログラムに実際に試してみた
    - 通常動作で1つの間違いが見つかった
    - 通信相手が異常なデータを送ったときに、3つのケースで正しく動作しないことも見つけた

実際の検査器はブースで展示しています

今回やったこと

- 具体的な取り組み (3)
  - さらに、プログラムの検証も取り組みました
    - 機械可読な仕様書と実際のプログラムから、動作が一致していることを厳密に証明しました
      - ただし、まだプログラムの一部分ですが
      - 証明が終わった部分は完璧に正しいはずです
    - この手の証明は非常に膨大になります
      - 普通の教科書風に印刷すると数千ページとか
      - 手でやると間違いの可能性が常にあります
      - そこで、「機械証明ツール」（定理証明支援系）というものを使います
    - こちらもポスターで展示しています

まとめ

- インターネット通信を支える
  通信ソフトウェアの安全性向上を目指し
  - 厳密な通信仕様を記述できる仕組み
  - 通信仕様とソフトの適合性を
    全自動・網羅検査する仕組み

  を開発しました
- 今後は…
  - より広範な応用で実験
  - 標準化などで世の中に貢献したいと思います