4. セキュリティとプライバシー

4.0 セキュリティ って ?

本当の 攻撃者は誰?

- 組織内部：大半
  - 社員は信用できない。
- 組織外部：少数
- 保守的人種 (組織の内外を問わず)
  - 新しいもの(e.g.,技術)には危険/リスクはつきもの。。。。。でも、現状を維持したがる。

本当の 攻撃者は誰?

【注意が 必要な 常套手段】

- **たくさんの 攻撃 に 遭遇 しています。**
  1. 攻撃者 を 発見 したいですよね。
  2. **攻撃から 守って あげますよ。**
  3. 僕を信頼して 中身を 見せれば安心ですよ。

- **「繋いでいない」から 大丈夫です。**
  1. だから、セキュリティー対策は 不要 です。
  2. **でも、繋がるかもしれませんよね。。。**
  3. **システムをアップデートすると 保証 できなくなってしまいますよ。 (脅し、恫喝)**

セキュリティー の「大枠」

- 何が問題なのだろう?
  - 「個人情報保護法」、「青少年ネット規制法」
  - 「組織犯罪処罰法」 (=「共謀罪法」、「テロ等準備罪法」
- 「(情報)セキュリティー」はどうあるべきか?
  - 安心してのびのび仕事ができるような環境

    (*) 事故が起こらないように 委縮した活動環境?

    (*) 実は、職場の「安全衛生管理」と同じ。
  - 「野性児/放蕩」と「箱入り」、どっちが強い?

    ➔ <u>意図的な“Diversity”環境の構築</u>

How do you think?

『項羽と劉邦』(司馬遼太郎著)より

。。。。やがて華何が死に、曹参は後任を命ぜられた。 彼は、斉の丞相の職を後任に譲るとき、

「それでは、斉の獄市を貴官にお渡しします」

と言った。 獄市とは商品の市場のことである。むろん、この時代といえども政治は多岐にわたっており、獄市のみではない。 後任者は不審に思い、政治にはほかにもっと大事なものがあるのではないでしょうか？ と反問すると、

「獄 と 市だけが、政治の要です」

と、曹参は言った。 曹参の考えは、牢獄も商業の場も、善悪ともに 受け容れるところです、これに対して**為政者が善悪に厳格でありすぎると、かえって ぐあいが悪くなります**、ということであった。

・・・(略)・・・・

曹参は、世の中には必ず姦人とい者がいる、という。 これをやわらかくつつむのが、曹参の社会に対する生理学的な認識のようであった。 そういう姦人たちは、司法の対象になるか、市場管理の対象になるかどちらかだが、この獄と市をあまり やかましく正しすぎると姦人は世に容れなくなり、必ず乱をおこし、国家そのものを毀損することになる、だから獄市は大切だと言ったのです、曹参は答えたと いう。

How do you think?

『項羽と劉邦』(司馬遼太郎著)より

禁酒法

麻薬

。。。。。が て 曹参は後任を命ぜられた。 彼は、斉の丞相の職を後任に譲ろ

「そ 渡しします。

と言った。 獄 ことである。 も政治は多岐にわたってお り、獄市のみではない。後任 台にはほかにもっと大事なものがあるのではないでしょうか！ と 尋ねると、

「獄 と 市だけが、政治の要です」

と、曹参は言った。 曹参の考えは、牢獄も商業の場も、善悪ともに 受け容れるところです、これに対して**為政者が善悪に厳格でありすぎると、かえって ぐあいが悪くなります**、ということであった。

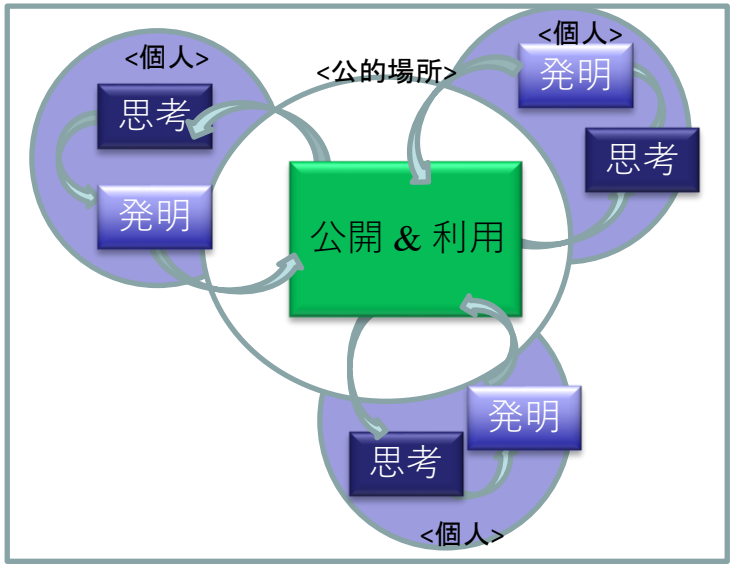
・・・(略)・・・・

曹参 む
のが、 は、
司法の
り やか の
ものを
と いう

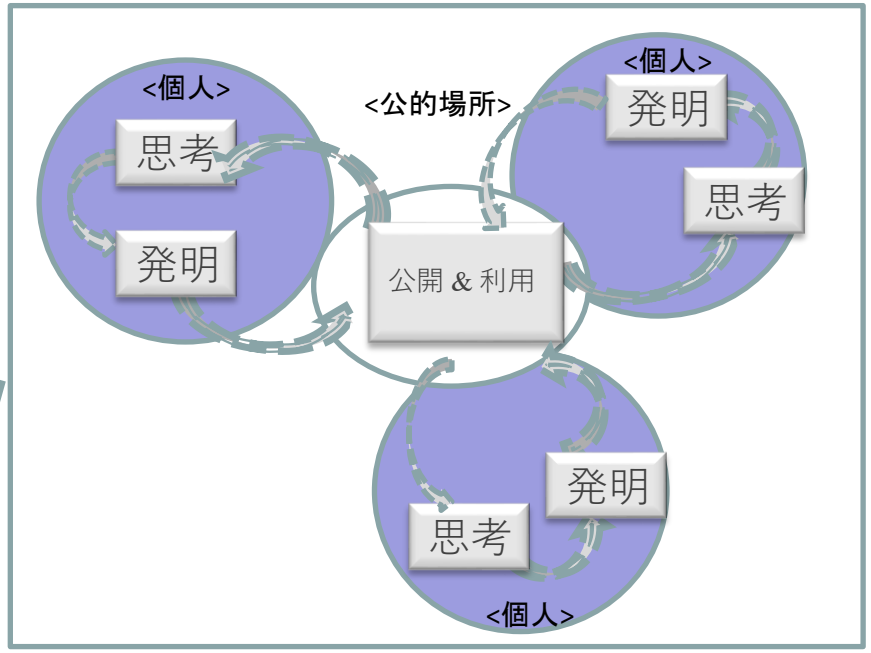
【麻薬】➔『非刑事問題化』

1. オランダ
2. ポルトガル(2001年)：病気として治療支援

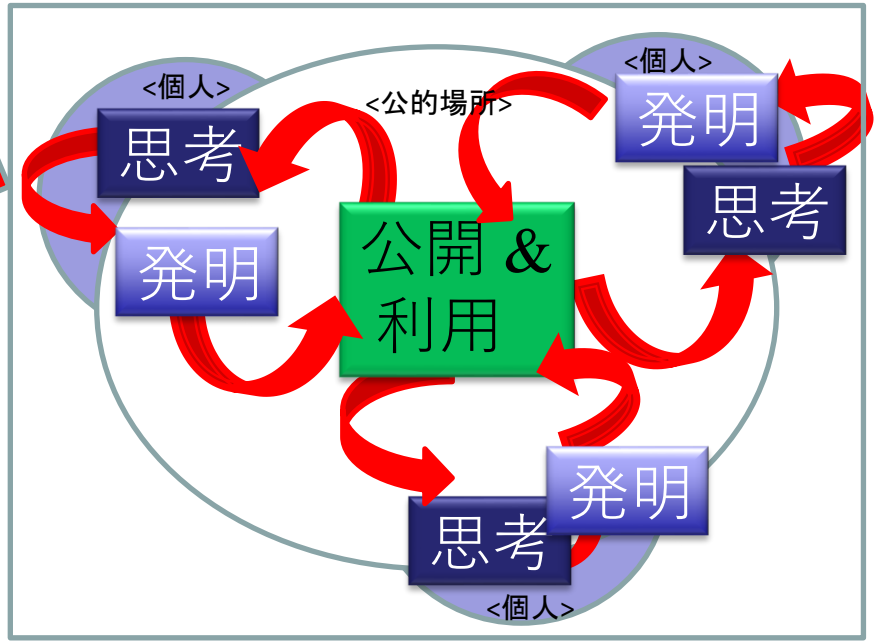
著作権・知的財産権の規制もしかり



過度な規制



交流の促進



“セキュリティーの(正しい)意義

【誤】: ① 風紀委員の増強
② 安全(“ゼロ”)の実現

【正】 1. “安心”の実現
2. “のびのびと”仕事
3. リスク対応(“non”-ゼロ)

セキュリティーと辞書で引くと

安心 と 安全
は 同じ なの？

- 安全, 無事
  - public ～ 治安, 公安/in ～ 安全に, 無事に.
- 安心, 心丈夫
- (財政上の)安定, 保障:⇒social security.
- 〔危険･危害などに対する〕防衛(手段), 警備(態勢), 安全保障〔against, for〕

- (負債の支払いに対する)保証, 担保, 抵当(物件)、保証人、有価証券:government securities 政府発行の有価証券(e.g., 国債･公債)

安心

- 個人の**主観的**な判断に大きく依存するものである。当懇談会では安心について、人が知識・経験を通じて予測している状況と大きく異なる状況にならないと信じていること、自分が予想していないことは起きないと信じ何かあったとしても受容できると信じていること、といった見方が挙げられた。人々の安心を得るための前提として、安全の確保に関わる組織と人々の間に信頼を醸成することが必要である。互いの信頼がなければ、安全を確保し、さらにそのことをいくら伝えたとしても相手が安心することは困難だからである。よって、安心とは、安全・安心に関係する者の間で、**社会的に合意されるレベルの安全を確保しつつ、信頼が築かれる状態である。**完全に安心した状態は逆に油断を招き、いざというときの危険性が高いと考えられる。よって、**人々が完全に安心する状態ではなく、安全についてよく理解し、いざというときの心構えを忘れず、それが保たれている状態こそ、安心が実現しているといえる。**

2004年4月 文部科学省
「安全・安心な社会の構築に資する科学技術政策に関する懇談会」の報告書

安全

- 人とその共同体への損傷、ならびに人、組織、公共の所有物に損害がないと**「客観的に」**判断されることである。ここでいう所有物には、無形のものも含む。世の中で起こりうる全ての出来事を人間が想定することは不可能であり、安全が想定外の出来事により脅かされる可能性は常に残されている。そこで、リスクを社会が受容可能なレベルまで極小化している状態を安全であるとする。同時に、社会とのコミュニケーションを継続的に行う努力をすることにより、情勢に応じて変動しうる社会のリスク受容レベルに対応する必要がある。**安全を高めようとすればするほど、利便性や経済的利益、個人の行動の自由等が制約され、プライバシーが損なわれる可能性がある。**よって、安全性を向上させる際には、このようなトレードオフの関係を考慮する必要がある。しかしながら、より高いレベルの安全を実現するためには、安全と自由のトレードオフの次元にとどまらず、安全性と行動の自由やプライバシーを並立させる努力を続けることが重要となってくる。

2004年4月 文部科学省
「安全・安心な社会の構築に資する科学技術政策に関する懇談会」の報告書

セキュリティー

- 経済面での観察

1. (とても) お金がかかる

   完全性が保証できない。。。。

   ➔ お金を払いたがらない

   事故が起こらないと必要性を認識(実感)しない。

   ➔ 適度な事故が発生しないとお金を払わない。

- ということで、セキュリティーのビジネスが成立する社会 ＝ “適度に” 事故がある

➔ これが、Post Covid-19 で変わるかも！

セキュリティー

- 必要なこと / 実現すべきこと

  守るべき物に関して、以下の2つを実現。

1. **破壊・変更(改竄)**されない

   再生コストを必要とする事故

2. **盗難**されない & **不正利用**されない

   a. 再生コストを必要する 事故

   b. 他人のコスト負担を必要とする事故

セキュリティー

- 例えば、デブ への生命保険
  - 困ること
    - 死亡 ➔ 収入の喪失
    - 疾病発生➔治療費

    (*) つまりは、収入の減少
  - 保障すべき物
    - お金
  - **最近保険会社がやっていること**
    - **デブにならないようにする方法を奨める**

      i.e., 予防策の伝授 ➔ 保険利用の確率を下げる。

セキュリティー

- 可能な対策
  1. (修復)コストを保障/補償 (保険, insurance)
  2. 事故を未然に防ぐ (Pro-active)
  3. 事故発生時の対応策
     a. 事前策 (Pro-active)
     b. 事後策 (Re-active)

- 注意すべき点
  1. 完全性は要求不可能

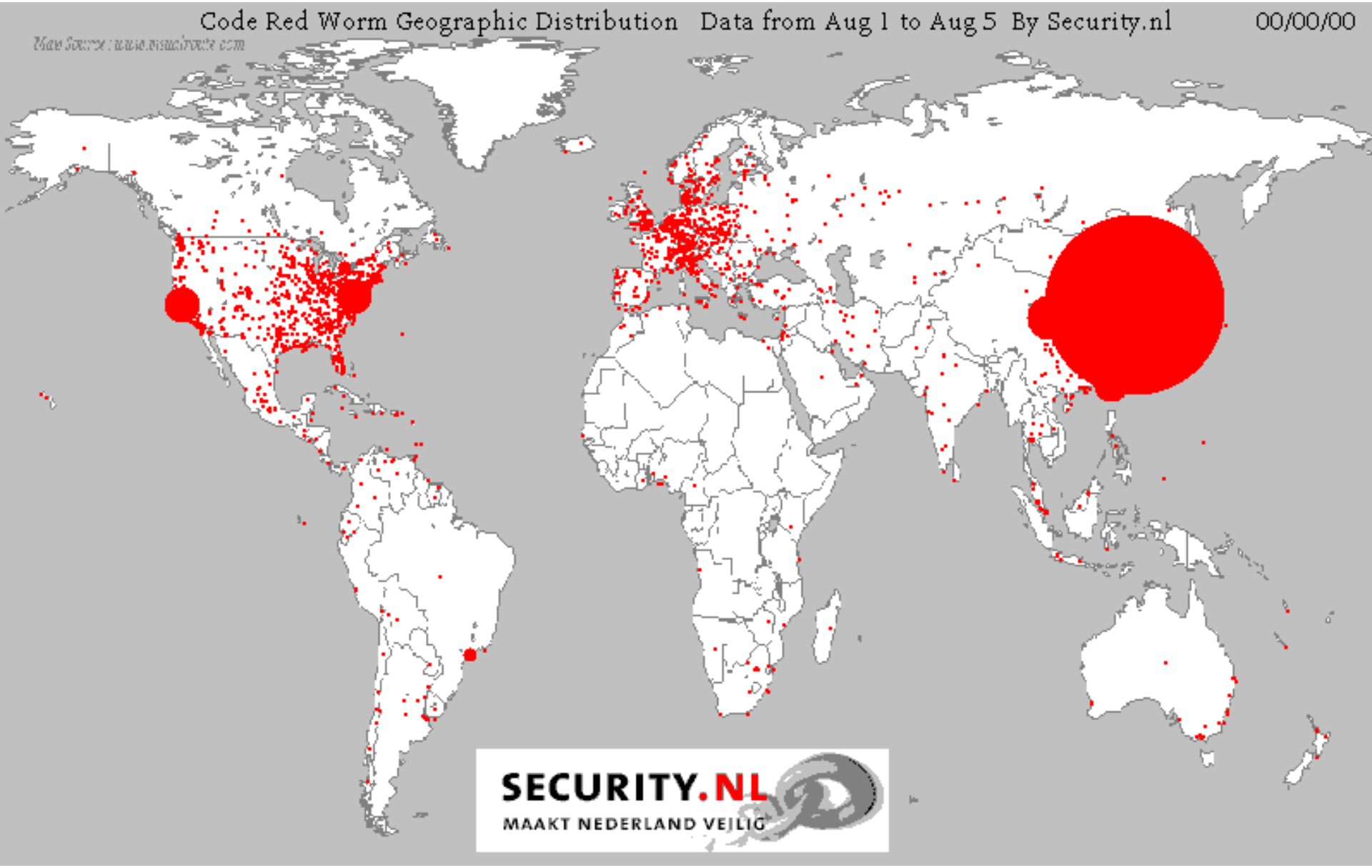
     (*) 事前、事後の両面において
  2. “完全な” 社会ではセキュリティービジネス自体が成立しない

いろいろなモデルがある。。。。

- SLAMERの例からのレッスン
  - 先進国(e.g., 北米/欧米/日本)
    - 合法ライセンスが90%
  - 発展途上国(e.g., 韓国)
    - 違法ライセンスが90%

      (*) Windows Update Patchなし。。。。。
  - 後進国 (e.g., 中国)
    - 徹底した違法ライセンス

      (*) Windows Update Patch込みの違法コピー


Code Red Worm Geographic Distribution Data from Aug 1 to Aug 5 By Security.nl
00/00/00
SECURITY.NL
MAAKT NEDERLAND VEILIG

いろいろなモデルがある。。。。

- オープンソース至上主義
  - すべての ソフトウェアは、オープンソースでなければならない。
  - ブラックボックス VS オープンソース

    (Microsoft)　　(Linux)

  - ブラックボックスがないとセキュリティーは実現不可能
  - ブラックボックスだと、手が出せない
  - **オープンソースは、誰でも設計図を持ってしまう。**
    - **でも、みんなで、守る方法を考え共有できる。**

ところで、コストはどうなる？

- 手段の共通化共有化が容易
  - 陽: 守りの強さを常時同レベルに維持可能
  - 陰: 攻めるための情報
- 鍵 の複製コストが低下
  - 陽: いつでも、鍵は変えられる
  - 陰: 鍵は すぐに 複製され流通可能

セキュリティーの経済性(まとめ)

1. BCP は『セキュリティー』
2. 『セキュリティー』は 、常時は邪魔者&不要。
3. 無事故が続くと、『さぼりたくなる』
4. 『さぼっても』、利益構造には変化がない。。

What we need ?
What we offer you

【BCP 実施 が、ビジネスに貢献】

e.g., **効率化の情報提供**

プライバシー

“Privacy by Design” (PbD)
7つの原則

1. リアクティブ(事後)でなくプロアクティブ(事前)
2. ディフォルト設定でプライバシー保護
3. 設定時に組み込むプライバシー対策
4. ゼロサム ではなく ポジティブサム
5. エンド・ツー・エンド でのセキュリティー
6. 可視化と透明性 (オープン性・公開性)
7. 個人のプライバシー尊重(個人を主体に)

プライバシー

- (他人からの干渉を受けない個人の)私生活
- 秘密, 内密
- 隠遁(いんとん)
- Private の名詞形

かなり、主観的な面が強い。。。。。
➔ 結局は、セクハラと 類似している

プライバシー

- 結局は、セクハラ、アカハラ、ドクハラ と似ている。
- 同じ事象でも、問題がない場合と、問題になる場合とがある。
- 基準も時間ともに 変化していく

- 必ず 騒ぎ出す ひとがいる。
  - 最初から話をするべし！

プライバシー

- 対策は、責任の所在を明らかにすること
- 基本的には、個人で守るしかない。
- しかし、個人情報に 合った サービスを 顧客はありがたがる。

  ➔ 情報の 2次流通、2次利用に関するコンセンサスの必要性

プライバシー

- 結局は、誰が(Who)、何の目的で(Why)、どのように(How & Where)、いつ(When) 個人の情報を利用するか。
  - ➔ なんだ、5W 1H じゃん。
- デジタルもアナログも基本的には同じ。デジタルは、情報の流通速度を大きくする。

プライバシー

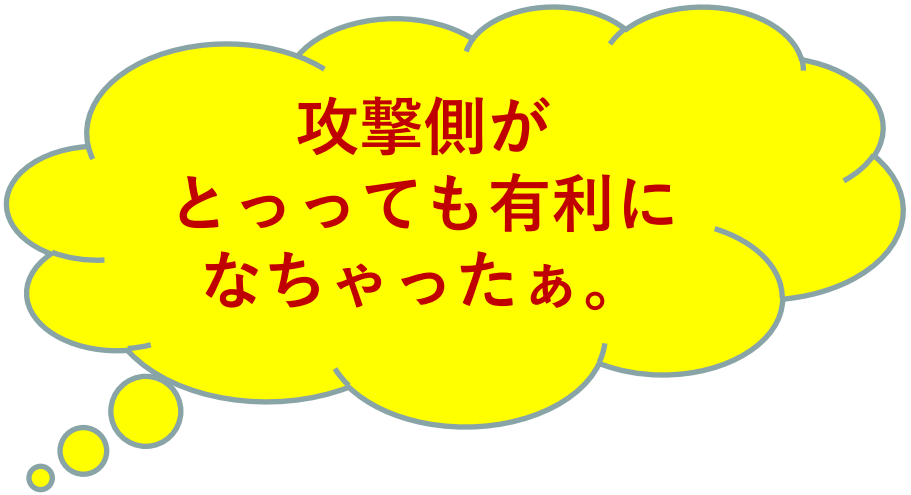
- ところが、、、、、
  - 情報理論：

    秘密の情報(＝貴重な情報) ほど、高い価値がある。
  - プライバシー

    公然の事実（＝普通の情報）になると、プライバシーではなくなる

プライバシー

- デジタル技術の効果
  - 複製
  - 流通
  - 共有
  - 加工



が **簡単に できちゃう !!!**

ところで、通信の秘匿性

- 何のために、秘匿性が必要なのだろう？
- 「匿名性」は 何の ために必要なのだろう？

- なぜ、「ブロッキング」は、大きな 議論の対象になるのだろう？

セキュリティに対する考え方 (案)

1. **グローバル**に考え、**ローカル**な施策を行う
2. 「原理主義」ではなく**「実践主義」**で進める
3. 強制する・制限するのではなく、**活動の活力向上**を応援する
4. **「過保護」**は、かえってリスクを増大させる
5. 「やらされる」ではなく、**「やりたくなる」**を目指す
6. セキュリティ対策を、**品質向上のための投資**と捉える
7. 経験と知見の**「共有」**を行う
8. インシデントの経験者は、「被害者」として**「保護・支援」**する
9. **「匿名性」の堅持** と プライバシーの保護
10. **まずは自助、次に共助、最後に公助**

4.1 暗号技術

暗号方式の分類

- 共通鍵（対称鍵、秘密鍵）暗号
  - 換字式暗号
  - 転置式暗号
  - プロダクト暗号
- 公開鍵（非対称鍵）暗号

資料提供：
TORUS 代表取締役社長 木村 幹夫 氏

インターネットの裏方
公開鍵方式

TORUS

私達が仕事でネットを使えるのは
重要なデータのやりとりができるから

重要な文書を安心して注1送れるのは
他人にのぞき見られたり
改ざんされたりしないから

注1：安全ではないが安心

TORUS

暗号化技術が
インターネットを支えている

どんなハイテクが使われているだろう？

TORUS

その前にちょいと歴史を

TORUS

人類は古代〜１９７０年代まで
暗号のときは一種類の鍵でやりくりしていた

TORUS

「暗号化が大変なら
複合化も同じくらい大変だよね！」
という発想

TORUS

できるだけ難しい暗号化を考えよう

でも
暗号の鍵と復号の鍵は同じ

どんなに難しい暗号の仕組みを作っても・・

それって

1.送受信の間に鍵を盗まれたらアウトだよね

2. たくさん通信したら鍵がばれやすくなる 注2

注2: 鍵を毎回変えればいい。毎回違う新しい鍵(Key-2)を鍵(Key-1)で知らせれば、はるかに ばれにくくなる。

TORUS

暗号以前に
鍵をどうやって安全に受け渡しするか？
の根本問題があった

何とかならないのか？

TORUS

鍵を分けたら良いんじゃないか？

TORUS

「暗号化だけ」できる鍵
&
「複合化だけ」できる鍵

TORUS

書類を送ってもらう相手に
「この鍵で暗号化してね」
とメッセージする

TORUS

暗号化した文書が送られてくる

解読できるのは
複合化の鍵が分かっている自分だけ

TORUS

復号の鍵は送らないので、
文書と暗号鍵が盗まれても、解読されない

TORUS

なかなかCOOL！

でも、暗号化の鍵から
複合化の鍵が類推できるなら
意味がないよね？

TORUS

暗号化できる鍵を知っていても
複合化できる鍵が作れなかったら良いはず

そんな都合の良い話があるのか？

TORUS

実はあった！

TORUS

これは何と何のかけ算でしょう？

65903455923832285609573592995818390001092749061041975023348216533577610995176130640367526195179841942093509410963954149527509952291913377016848693318906814699012263857868129972476578656510990553881111770258797585062261412672749948798378684242399702175669529502579664947866767640414810398762685894147335636178858623613567219655261002060926595514905342663211450670589508347104787010186858355630781484366632790369977861791556335557869580017327372103875140385810460655044030147845572561006818436489600372045362641278307748967621757649716793607127934505521483562660305502309963888076930350701402560886118587281648134837016434816928387584508117909444519620291361953315564941839065903698573664980615465313228356506045226628690844289616409404902047153870357570926608886305369101757419970753453065490075267186190015713949569023134120195069556224421918840251752047137476642488289261842750842716747427383529023842769107104409539201621874543017935433181379867168752703712683141770310524994307667356988327004423732386199971777413394221463720977807998802523366723615609282666143427913434093078378117599039754843645916071464436281955016155687395
3

注3: RSA暗号

TORUS

こたえ[注3]

素数p

313991399371199131139799331911377147529895941991587879456361416793343797754289852575517
133312684269943695978946644516863648961536981354977375935673418795287369494189373478623
641239162919379269294319941871985794933399739235523691657154837889117834232678974449658
279117129522895488222612449716435651112797868118722475112367318718359954332756851152845
673554343833423958324129279242571543956244312159149656971499164148747227159798119915531
789396889314926554998567389189177184378411356887579966732519395769634484946484155736859
195773976485587598811713196922772648319742413259665798111566314845954551344321292792178
583218155711143611735499324729469232679643212644511755544726594454683193623626957711324
895114496128478896375157597659974246467315936911531792288239249136494329788845728831611
728857639343337449493221561738959339141347119138332653219119612984163669317356624631952
956188127648784846583361813646131913157456632928169513747231224138425962243343371145487
745954412587484837933238642278851955148574512595199969685612245439118737626399742196143
742577819117917319979999777371311371999793393

と素数q

209889366574405864861512642566102225938639 21

TORUS

この組み合わせを見つけるのは

計算機を使っても大変！
何年もかかる

桁数を増やせば、1000年でも解けない
問題を作るのは簡単だよね

TORUS

元ネタの２つ（pとq）を知っていれば、
かけ算（p*q）は一瞬
でも結果(p*q)から元ネタ(pとq)の推定は
膨大な時間を削られる

TORUS

かけ算の結果 = 暗号鍵

元ネタ = 復号鍵

になっていれば良い

TORUS

かけ算の結果 ＝ 暗号鍵～（公開鍵）

元ネタ ＝ 復号鍵～（秘密鍵）

TORUS

任意の素数を２つピックアップすれば良いので
鍵は即座にいくらでも作れる

素数は無限にある

TORUS

その鍵を解くのは膨大な時間がかかる

計算機を使っても追いつかない

だから公開鍵方式は安全[注4]

注4: 正確には安心。計算機の能力(e.g.,量子コンピューティング)が向上すると暗号が解読される確率が大きくなる。

教訓

- 暗号のアルゴリズムには平文と暗号文の対応関係がわからない程度の複雑性を持たせ、個々の通信文の秘密は鍵を頻繁に変えることで維持
- 暗号のアルゴリズムは公開し、脆弱性がないか皆で検討

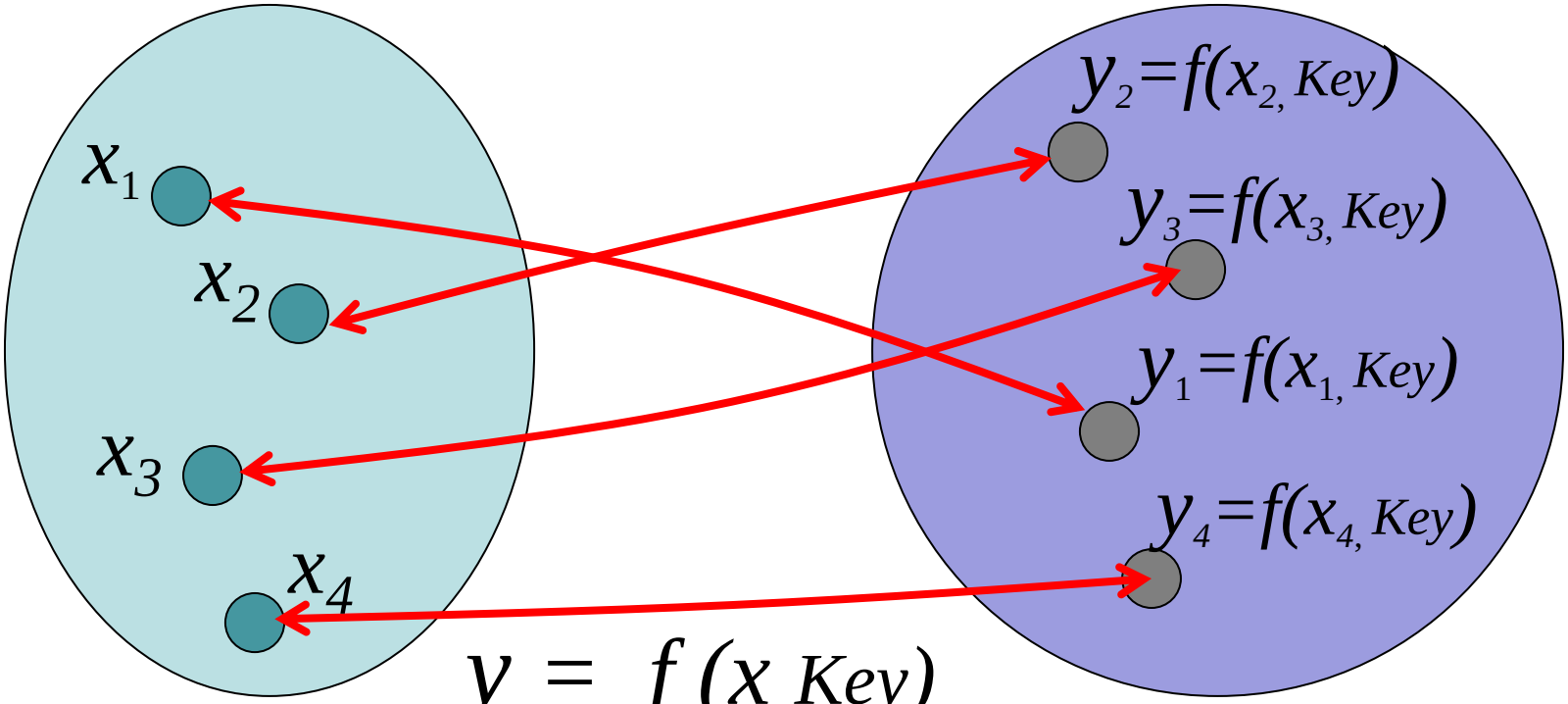
暗号化は何をしているのか。

- {平文}文字の空間 と {暗号}文字の空間との間の 写像{F(文字,パラメータ)} の計算を行っている。 {暗号}文を、たくさん 眺めると, 文字 と パラメータ が 見えてくる。
- この {たくさん} が、十分に大きければ よしとする。
- {暗号}文の空間 も、写像関数F もほぼ、無限に存在する。
  - 写像関数 F の 逆関数 F^{-1} が存在すること
  - 必ず、1対1 の写像となっていること。

(*) 認証では、この条件が少し甘くできる。

平文の シンボル
(e.g., 文字)集合(X)

暗号文の シンボル
(e.g., 文字)集合(Y)



$$y = f(x, Key)$$

$$x = f^{-1}(y, Key)$$

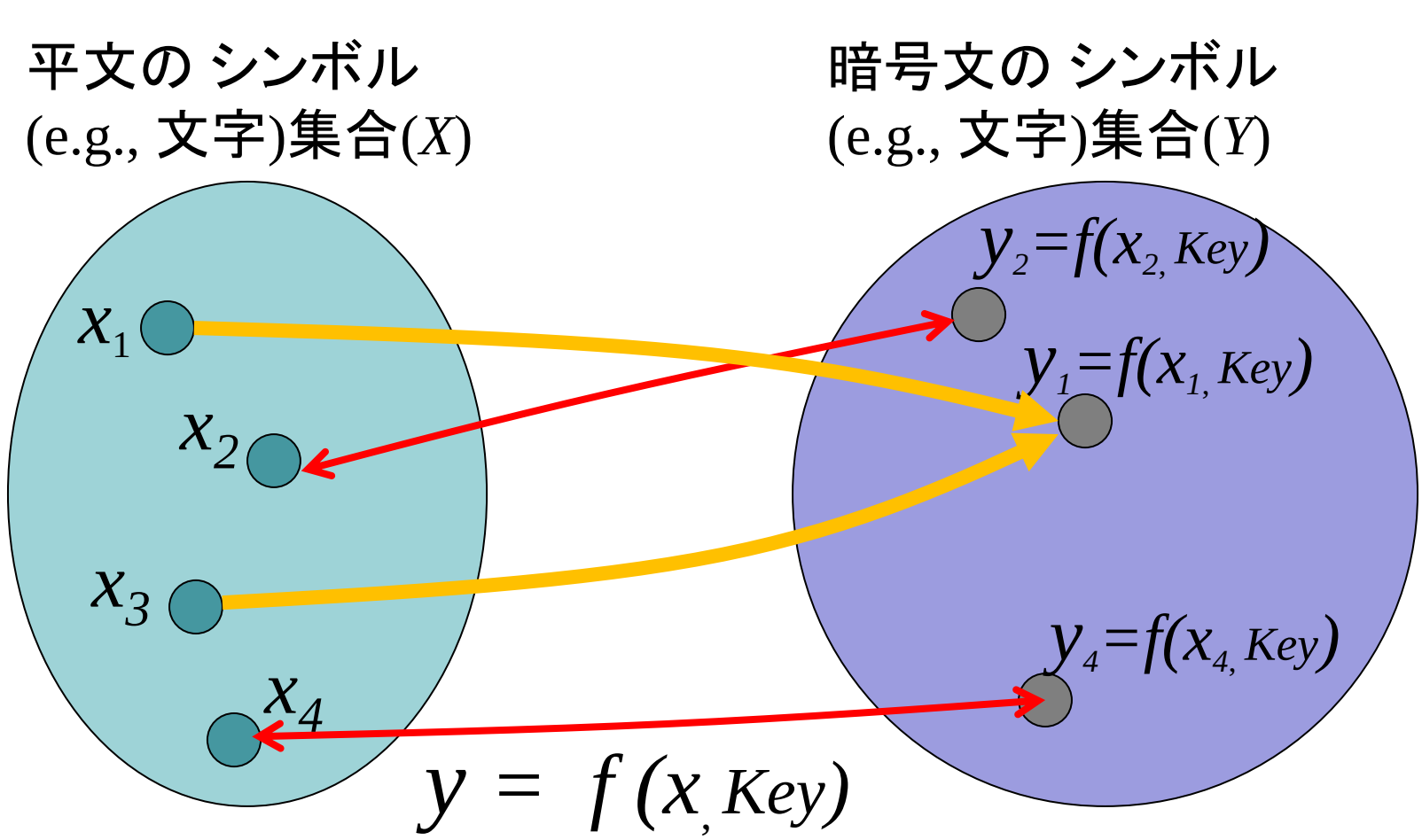
(*) 必ず、

(i) f^{-1} が存在

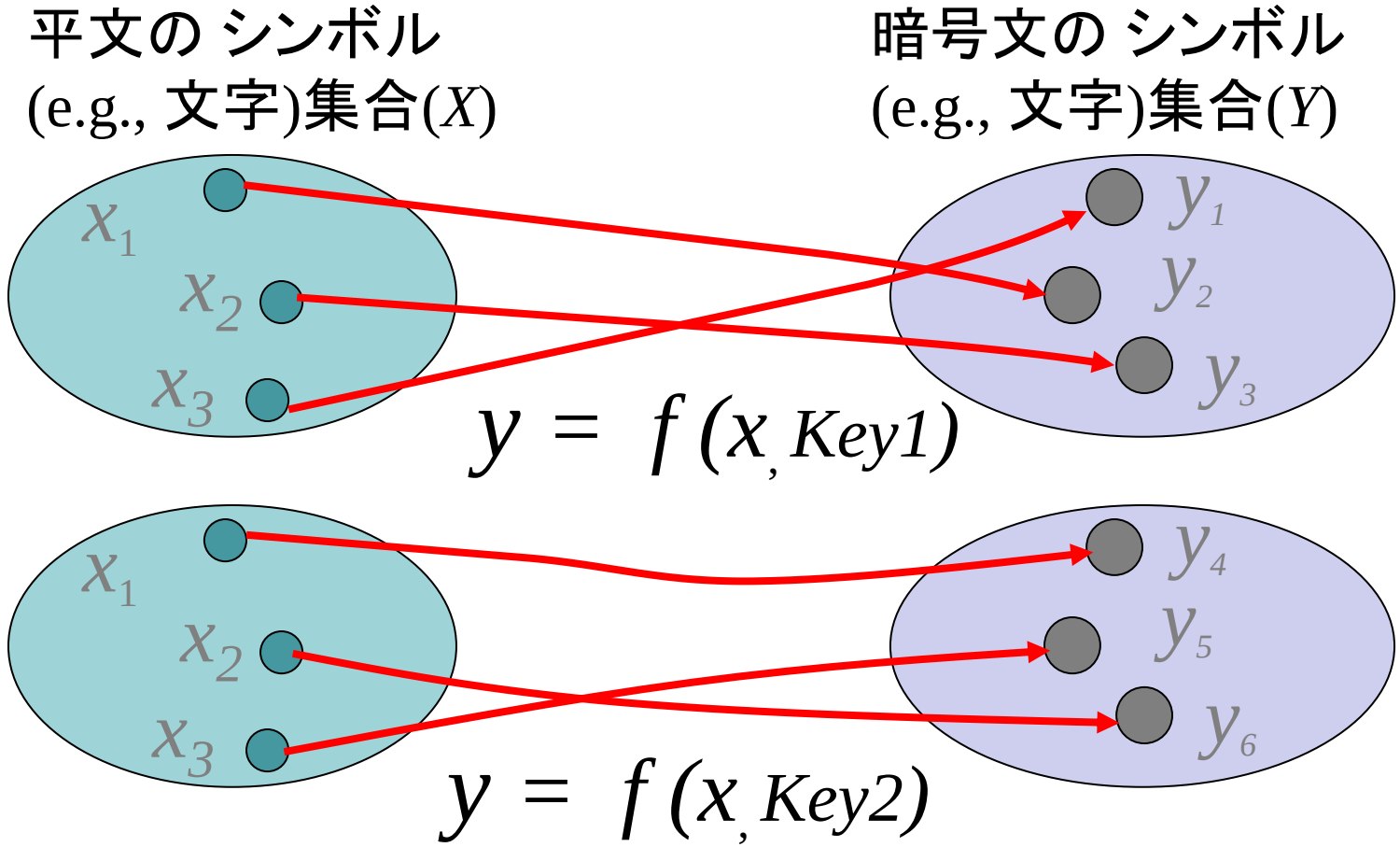
(ii) 写像は、1対1

$y_1 \neq y_2$ for any x_1, x_2

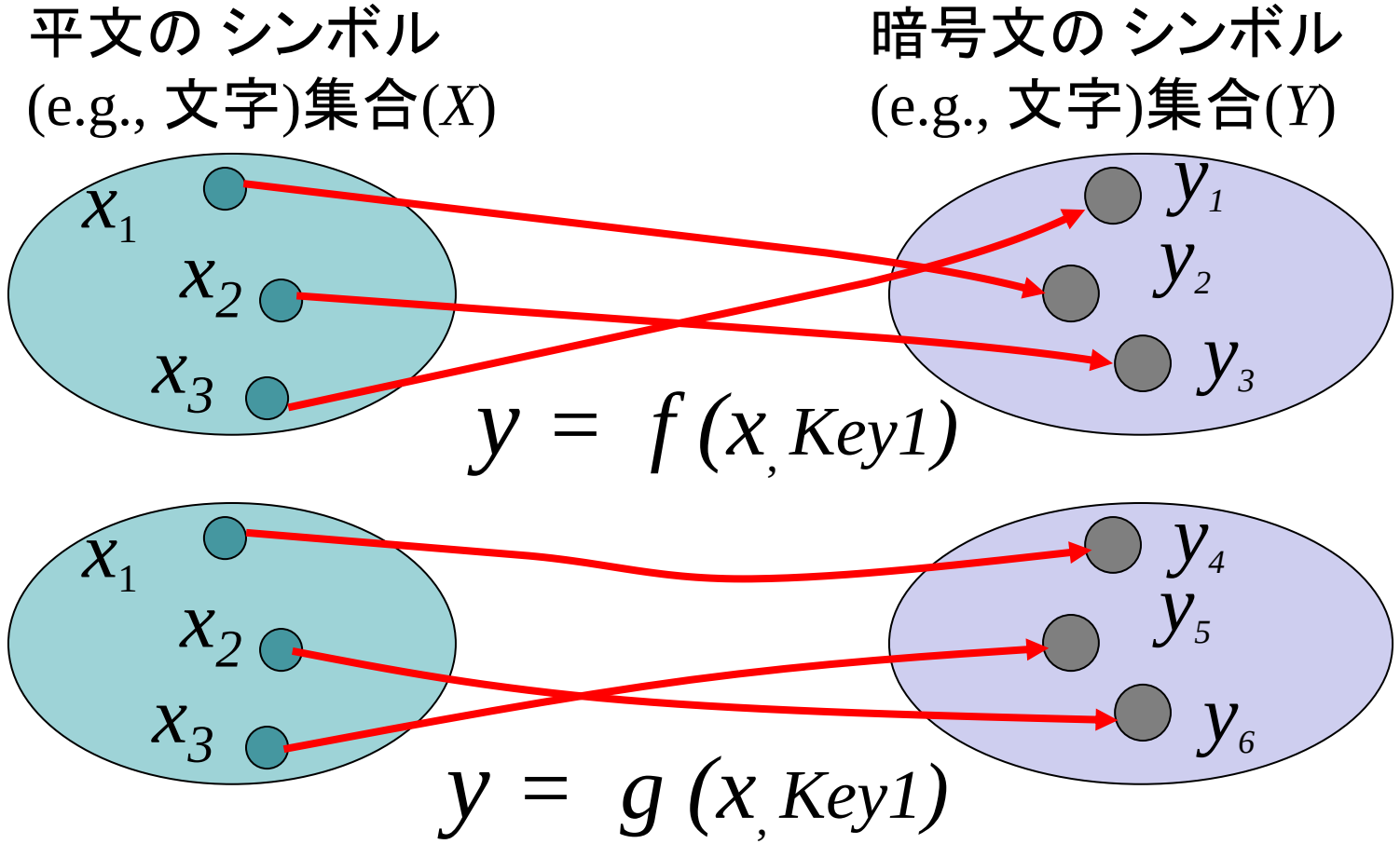
(*) *Key* は、ユーザごとに定義



(*) もし、同じ y に写像される x が複数存在すると
f^{-1} が存在しないことになり、暗号の 複合 が
できなくなる



(*) 同じ写像関数(暗号化関数) f を用いていても、
関数 f のパラメータである Key が異なれば、
写像値(暗号文のシンボル)も異なる。



(*) 仮に、暗号化に必要な 鍵 (key) が同じでも、異なる写像関数(暗号化関数) を用いることも可能。

標準暗号方式

- DES (Data Encryption Standard)
  - 64ビットブロック
  - 56ビット鍵
  - 16ステージ
- IDEA (International Data Encryption Algorithm)
  - 64ビットブロック
  - 128ビット鍵
- AES (Advanced Encryption Standard)
  - 128ビットブロック
  - 128ビットまたは256ビット鍵

公開鍵暗号

- $D_{k'}(E_k(P))=P$
- E_kから$D_{k'}$を推測することはきわめて難しい（選択平文攻撃によっても$D_{k'}$は破られない）
  - 例：大きな数の因数分解、離散対数、楕円曲線上の整数座標点

→E_kは公開してよい

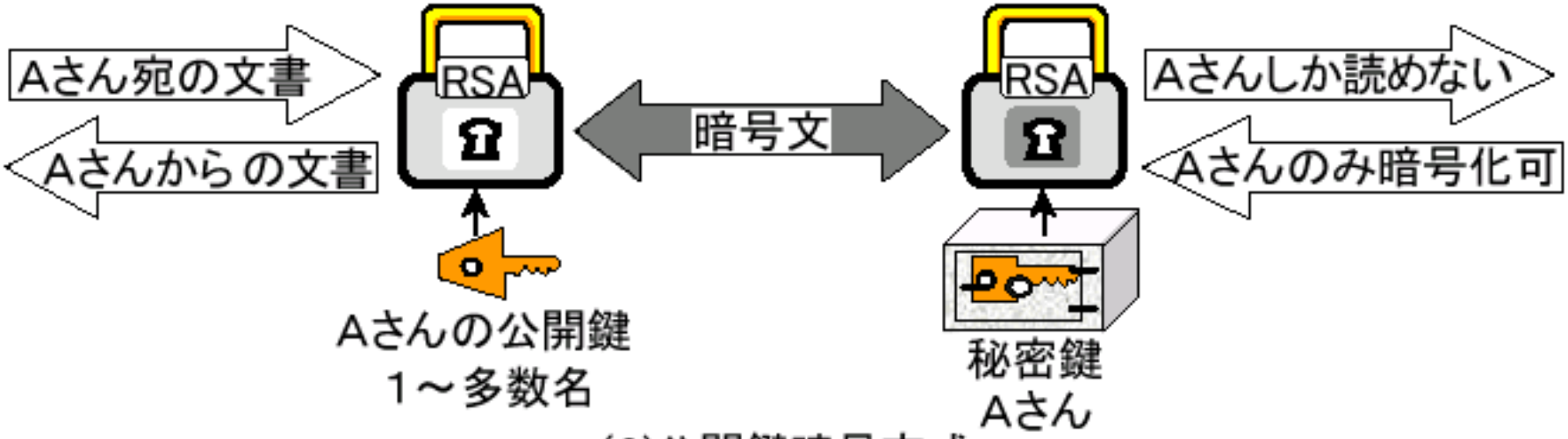
共通鍵暗号と公開鍵暗号の比較

- 共通鍵暗号
  - 必要な鍵の総数：$n(n-1)/2$
  - 各ユーザは$(n-1)$個の鍵を安全に保持する必要あり
  - 比較的高速
  - 任意のビットパタンを鍵として使用可能
- 公開鍵暗号
  - 必要な鍵の総数：$2n$
  - 各ユーザは自分の秘密鍵を安全に保持すればよい
  - 低速
  - 鍵として使用可能なビットパタンに制約有り

→鍵のビット数が同じなら共通鍵暗号の方が強い

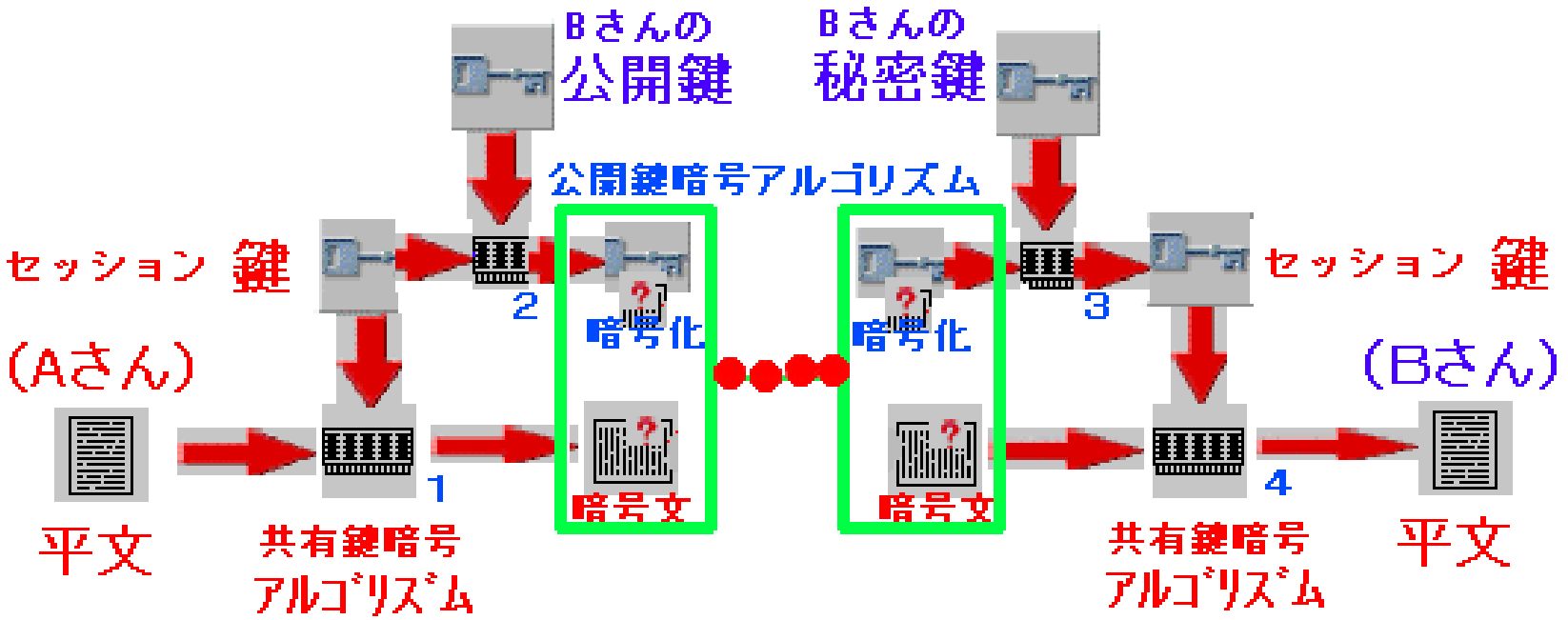
暗号方式の種類

(1)共通鍵暗号方式



(2)公開鍵暗号方式

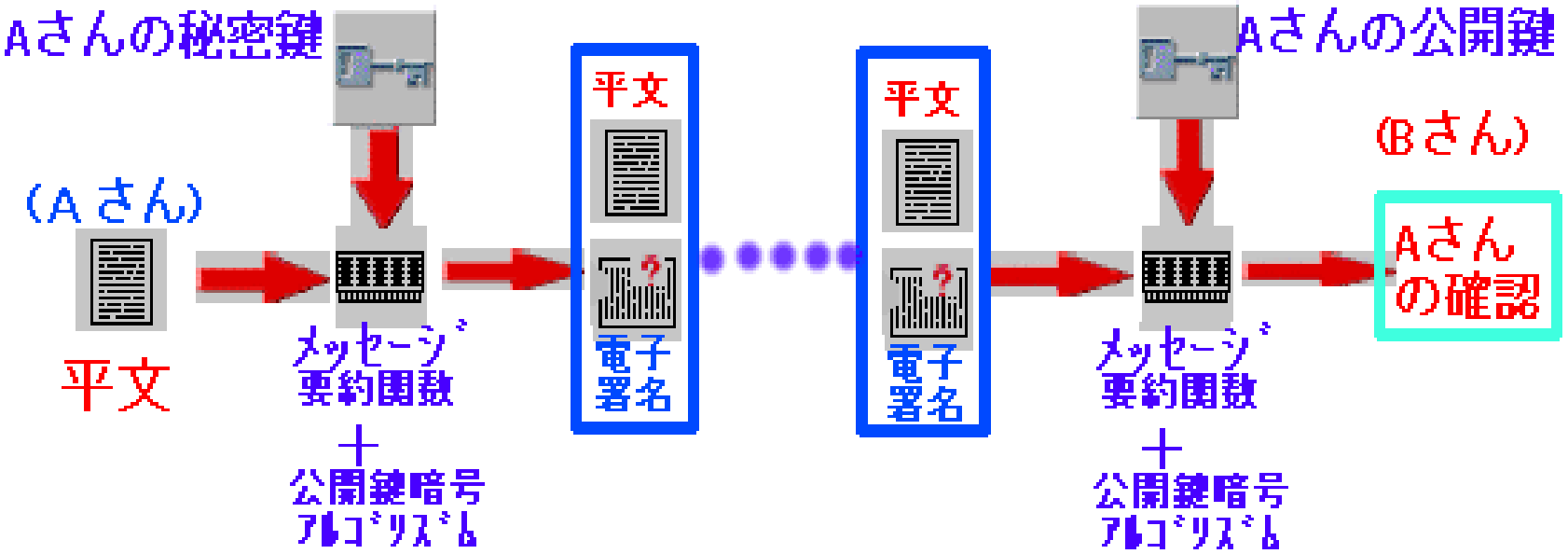
公開鍵暗号方式



例；ssh (Secured Shelll)

出典: http://www.psn.or.jp/TROUBLE/security.html

電子署名方式



- MD5 (128 bits)
- SHA (160 bits)

出典: http://www.psn.or.jp/TROUBLE/security.html

PGP署名メール

[署名したいメール]

７月３０日午前１０時に天満橋で会いましょう。

[署名されたメール]

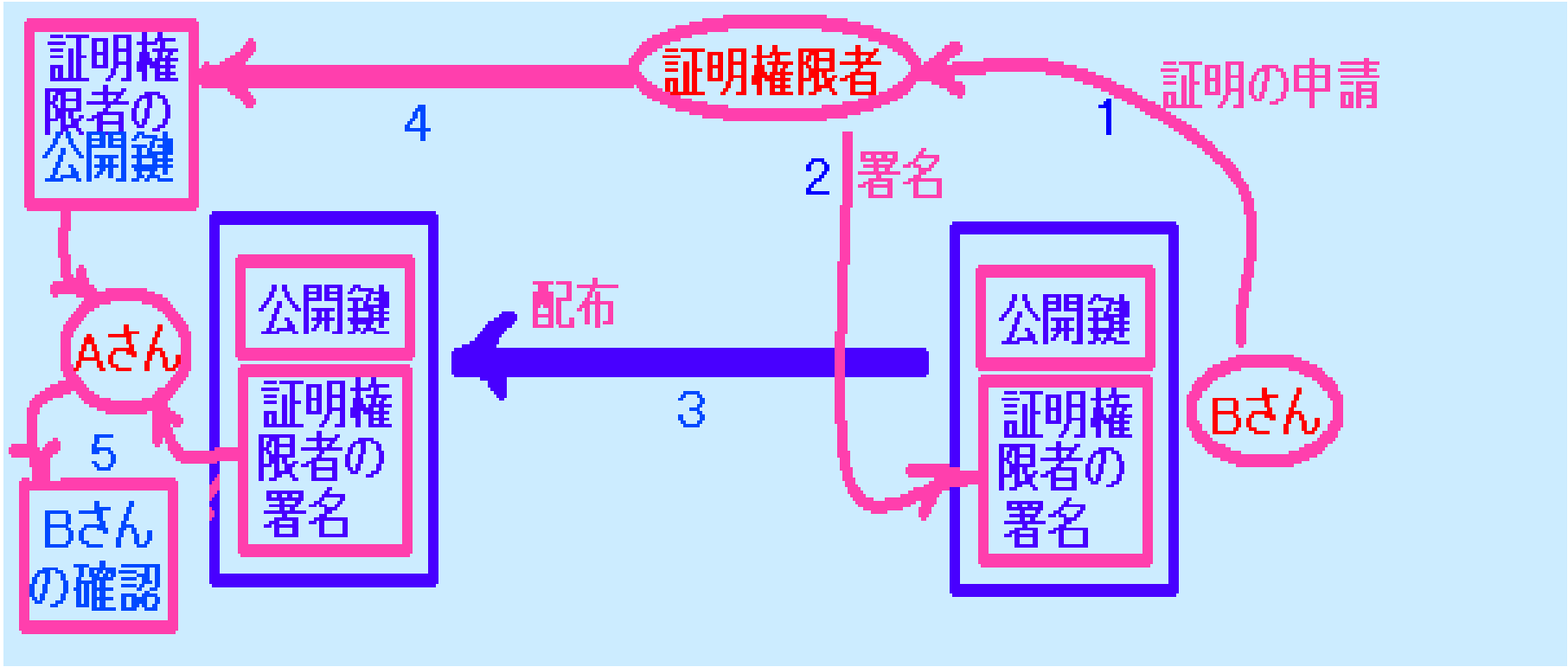
```
-----BEGIN PGP SIGNED MESSAGE-----

７月３０日午前１０時に天満橋で会いましょう。

-----BEGIN PGP SIGNATURE-----
Version: 2.6.3ia
Charset: noconv

iQCVAwUBMey8R6UtC+xzFETZAQEnUAP+N30di02slY+rRYa2gBJ2u2ImWofjeyks
1AkvsN9errDk4N/VcFmc3d6F4heDkiy87u3XAVoulz2orb9xZ3qFveoEZp3QLLa6
Pkzs6/N1nmJZFZFlf1M8yUR5WZTbyaVHQmC1AuSZhJsM8+8S/+IbpXVPJJ68M4JE
cDYBT86eekM=
=UE6f
-----END PGP SIGNATURE-----
```

出典: http://www.psn.or.jp/TROUBLE/security.html

鍵の証明と配布

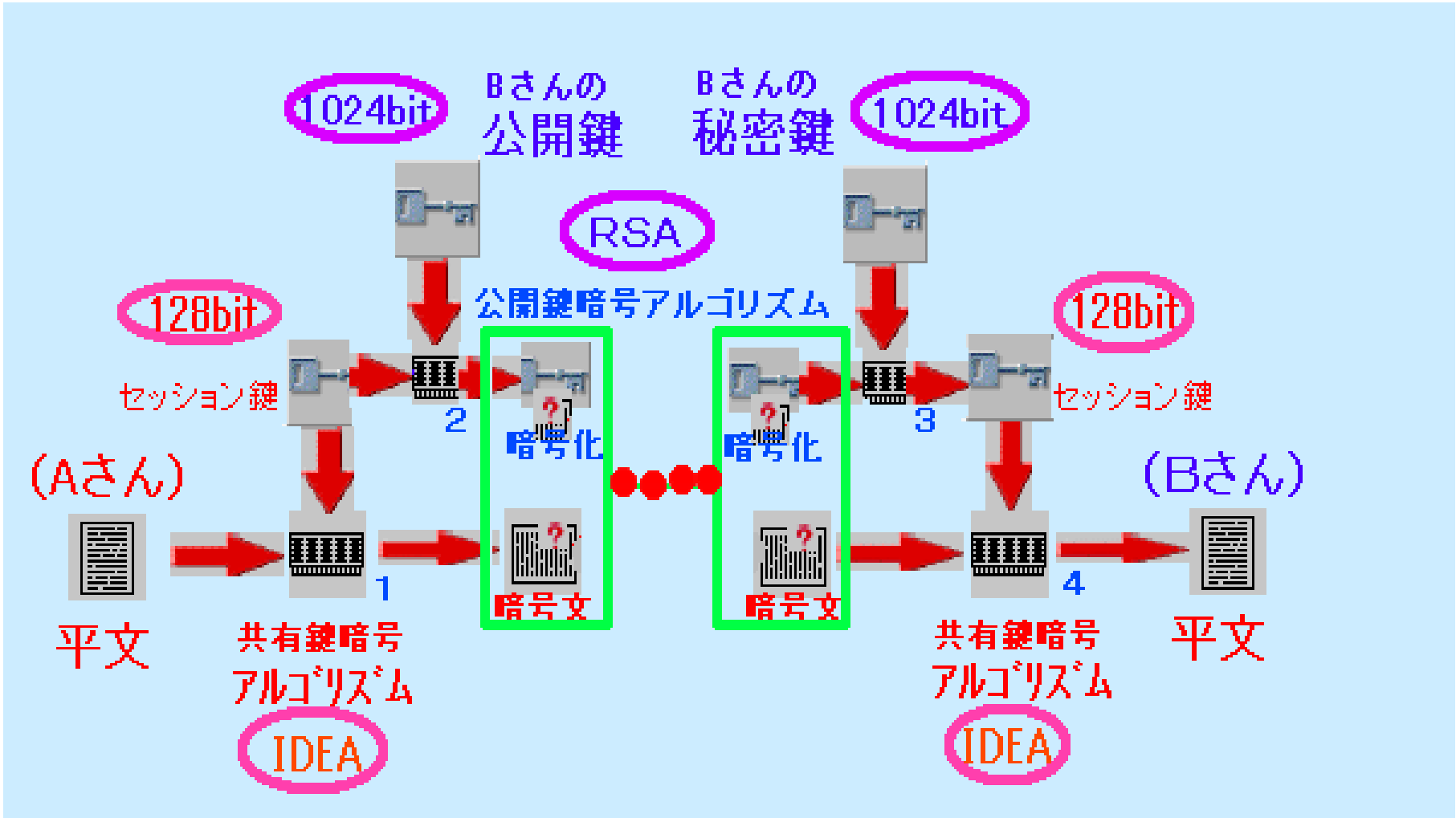
・証明を発行する機関・組織を利用

例；Netscape Webサーバ 暗号化情報交換



出典: http://www.psn.or.jp/TROUBLE/security.html

PGP暗号化メール



公開鍵暗号化方式(暗号化方式；RSA+IDEA)

出典: http://www.psn.or.jp/TROUBLE/security.html

PGP暗号化メール

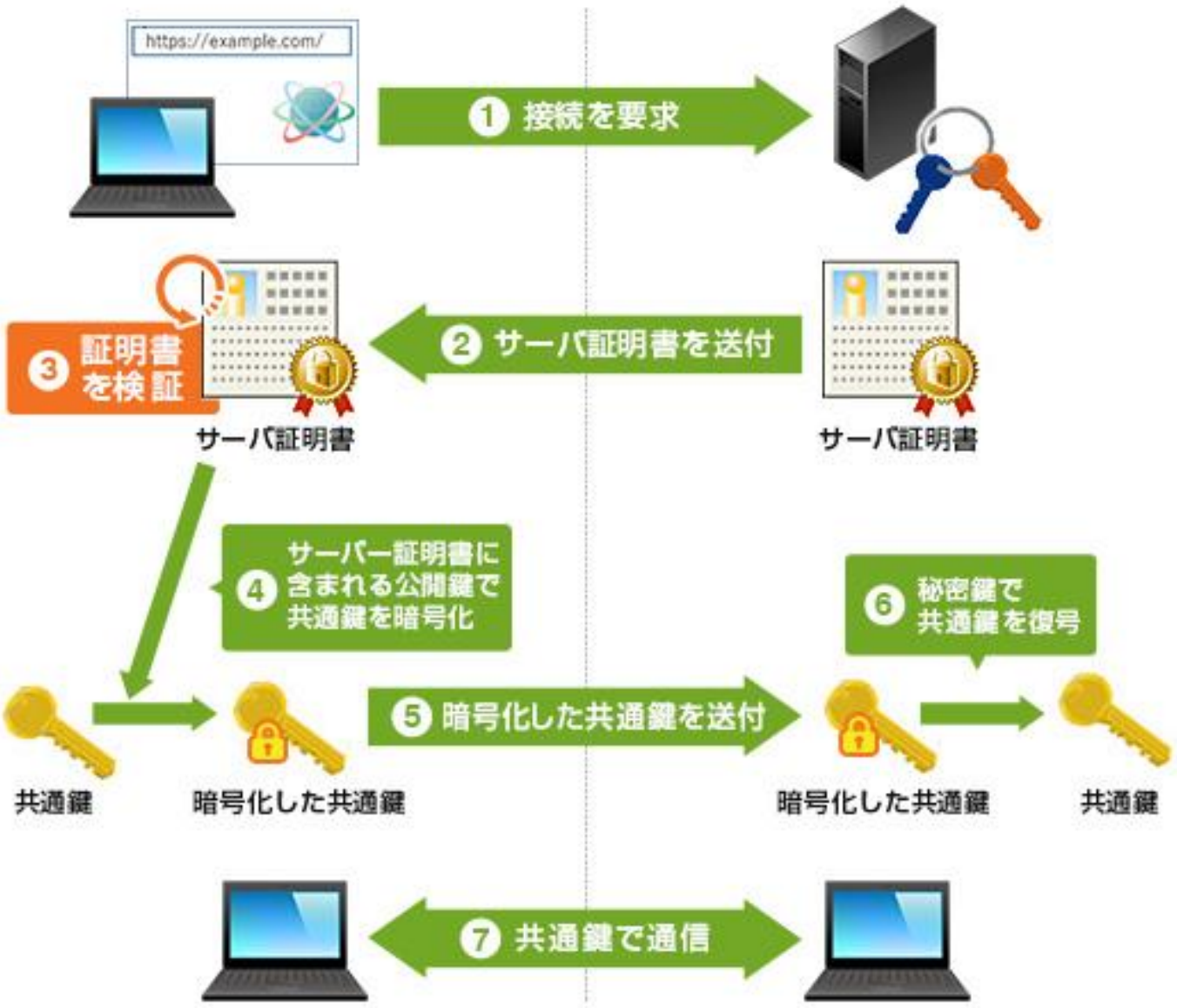
[暗号化したいメール]

７月３０日午前１０時に天満橋で会いましょう。

[暗号化されたメール]

```
-----BEGIN PGP MESSAGE-----
Version: 2.6.3ia
hIwDpS0L7HMURNkBA/4qk4BDXaiLag9tOS8srdd09IP4Pbocw8ERnYZKc8BJZHRq
bmePoSNRpv8QwRPttwB3pkUhPH9ET5BbGiyuw36hLvIet5z5ot3RS+XnfSz1Tyxw
xkXT+nNDCE6Gntb6JqBUym2/FRowwMNOc1bnKD6eIqZfekDUWBuHKSRduH6BfqYA
AAA3YBJcBDcrQtcIuA5R+bvivZ8gc8Fx3JCcUtW4yH+embVTTSUw+xTt0JSUoo93
 u5+LHGrrzBESSg==
=00WV
-----END PGP MESSAGE-----
```

出典: http://www.psn.or.jp/TROUBLE/security.html

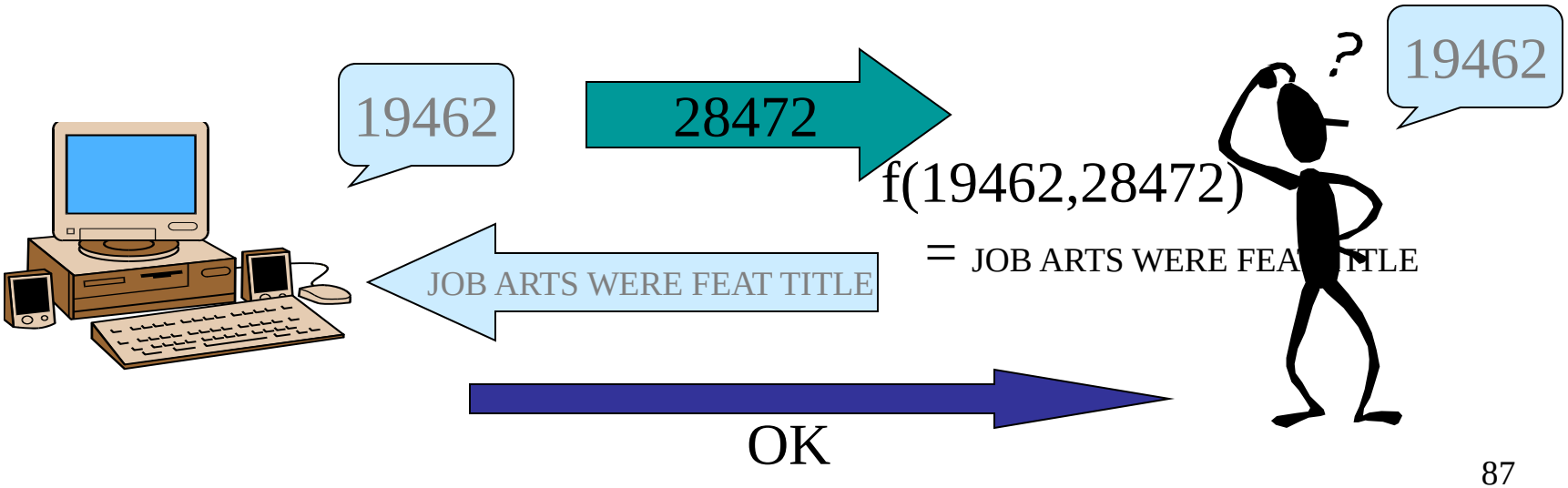

https://example.com/
1 接続を要求
2 サーバ証明書を送付
3 証明書を検証
サーバ証明書
サーバ証明書
4 サーバー証明書に含まれる公開鍵で共通鍵を暗号化
5 暗号化した共通鍵を送付
6 秘密鍵で共通鍵を復号
共通鍵
暗号化した共通鍵
暗号化した共通鍵
共通鍵
7 共通鍵で通信

4.2 セキュアなアクセス

使い捨てパスワード；OTP

- 毎回異なるパスワードを利用する。
- Challenge & Response 型の認証
- 同一の鍵をサーバーとクライアント間で共有する。
- 例えば、

  ftp://ftp.nrl.navy.mil/pub/security/opie/opie-2.3.tar.gz
- Windowsのクライアントソフトもある。



安全なRemote Shell
- ssh ; secure shell -

- 公開鍵暗号方式; 公開鍵＋秘密鍵
- フィンランド ヘルシンキ大学で開発
- RSA、IDEA、DES 方式による暗号化
- 例えば; http://www.cs.hut.fi/ssh
- Windows版クライアントもあります(Teraterm)
- sshd (secure shell daemon)
- ssh (rlogin)、scp (rcp)、ssh-keygen
- 鍵情報ファイル

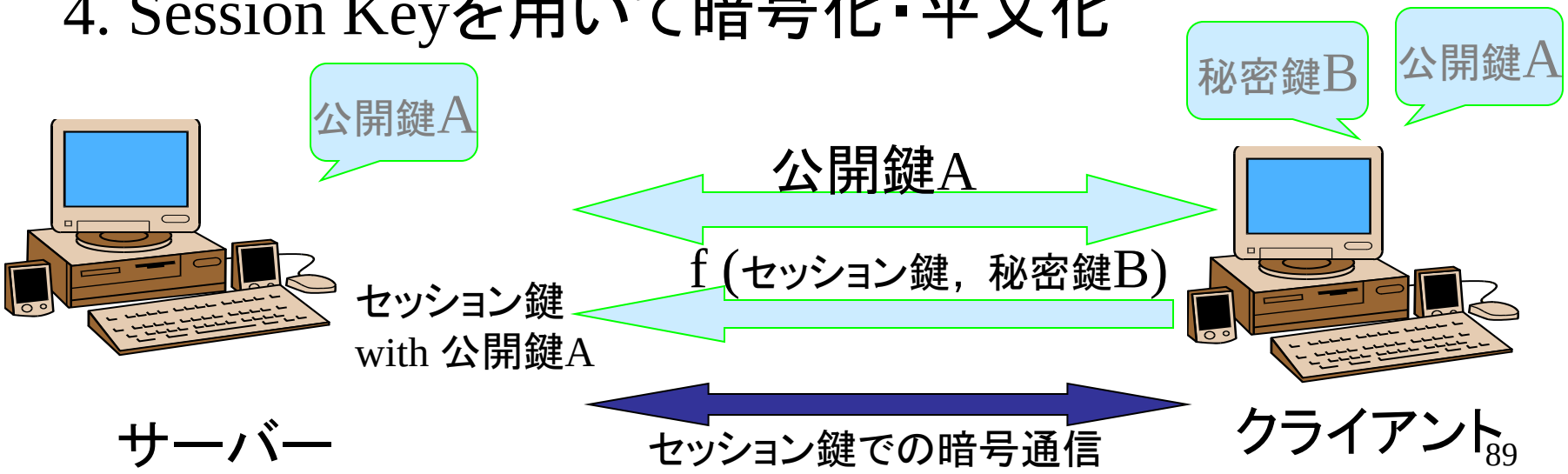
  .ssh/identity

  .ssh/identity.pub (client) => .ssh/authorized_keys (server)

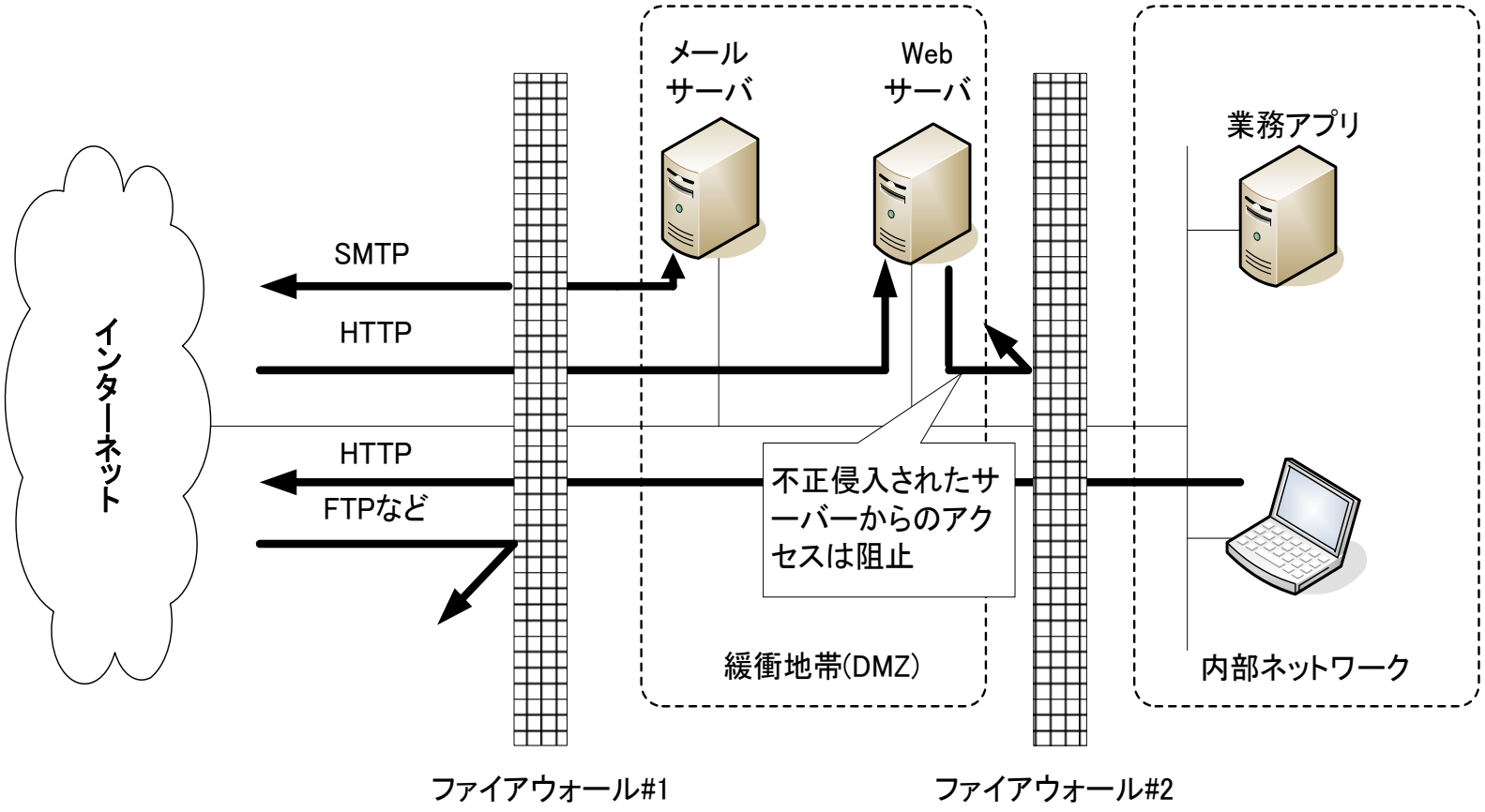
  .ssh/ssh_known_hosts, /etc/ssh_known_hosts

安全なRemote Shell
- ssh ； secure shell -

1. 公開鍵(Public key)の交換とチェック
   /etc/ssh_known_hosts、..ssh/known_hosts
2. クライアントは秘密鍵を使って Session Keyを送る
   (*) Session keyは毎回変わる。
3. サーバは公開鍵を使って Session Key を平文化する
4. Session Keyを用いて暗号化・平文化



ファイアウォールとDMZ



でも、、、、

- Firewall で、システムは 守れるのだろうか?

ゼロ トラスト ネットワーキングへ

クラウド技術 ＋ 携帯網技術

4.3 著作権保護

ガバナンスの講義の時に。

デジタルコンテンツと著作権

- デジタルコンテンツ：複製が容易かつ複製による劣化が生じない
- プログラムの不正コピー
- 海賊版CD
- Napster, WinMX, Winny等のピアツーピアファイル交換ソフトウェア

電子透かし

- コンテンツのヘッダ等ではなくコンテンツ本体に情報を埋め込む
- コンテンツの品質をできるだけ劣化させず、コンテンツを編集・加工しても埋め込んだ情報が消えないことが望ましい

情報の埋め込み・検出方法

- 人の知覚特性を利用し、人が知覚困難な範囲でディジタルデータを改変
  - オリジナルデータと比較することで検出な方式
  - 鍵を知るものだけが検出可能な方式
  - 誰でも検出可能な方式

電子透かしの用法

- 権利保有者の情報を埋め込む
  - 不正使用を見つけた際に権利を主張
- 一次取得者の情報を埋め込む
  - 不正使用を見つけた際に誰が「横流し」したのか確認できる

OSCON 2002

< Free Culture >

by Lawrence Lessig (Stanford Univ.)

July 24, 2002

1. Creativity and Innovation always builds on the past (創造とイノベーションは常に過去の上に築かれる)
2. The past always tries to control the creativity that builds on it (過去は常にその上に創造されたものと支配しようとする)
3. Free societies enable the future by limiting the past (自由な社会はこの過去の力を制限することで未来を可能にする)
4. Ours is less and less a free society (我々の社会は、日々自由を失っていく)

- 1774 Born the “Free culture”
  - Donaldson versus Backett trial ➔ stop copyright

    “Shakespeare is free”
  - 1710 Statute of Anne limited term 14 years
  - 1740s Scottish publishers reprint classis

    ➔ London publishers said “copyright is forever”

    “old patentees and monopolizers in the trade of bookselling, men who do not labour in an honest profession to learning is indetted”.

- 1790 in USA
  - Unregulated creativity,

    exception was “printing”, regulated for 14 years

    著作権法は、印刷物のみに適用され、派生作品には適用されず。14年間印刷物にのみ適用。
- 1928 Walt Disney……
  - Steamboat Willie (Micky Mouse) stole (i.e., rip, mix, burn) Buster Keaton “Steamboat Bill Jr.”
  - “Always parroting the feature length mainstream films”

- ピノキオ
- シンデレラ
- 不思議の国のアリス
- 海底2万マイル
- 眠れる森の美女
- 海賊船
- ジャングルブック
- リトルマーメード
- 美女と野獣
- ノートルダムの鐘

Brothers Grim (グリム童話) was free code as commons

- 1790 : 14 years
- 1804 : 28 years
- 1831 : 42 years
- 1909 : 56 years
- 1962 : 59 years
- 1965 : 61 years
- 1967 : 63 years
- 1968 : 64 years
- 1967 : 63 years
- 1968 : 64 years
- 1969 : 65 years
- 1970 : 66 years
- 1971 : 67 years
- 1972 : 68 years
- 1973 : 69 years
- 1974 : 70 years
-
-
-
- 1998 : 95 years

- No one can do to Disney, Inc. what Walt Disney did the Brothers Grim…..

2000s……

- Regulations
  - "publishing" to "copying"
  - "copies" to "derivative works"
  - "14 years" to "life + 70 years"
  - "opaque" creativity…….

    == can protect from exposure of proprietary intell's .

(*) Fair-use vs unregulated (i.e., free)

Unregulated…

- Read
- Give
- Sell
- Sleep

- Regulation
  - Only on ”publishing”

- Three camps;
  - Unregulated
  - Regulated, aka fair use
  - Copyright, aka protected

Proprietary

vs

Free/Open

Bill Gates (1)

- 今日使われているアイデアを考案した人々が、特許はどのように許可されているかを知り、特許を取得していたとしたら、今頃この業界は、完全に行き詰まっていたに違いない。

Bill Gates (2)

- 我々取るべき戦略は、所得できる限りの特許を取得することだ。独自の特許を持たない将来の新進企業は、先行する巨人たちの課すどんな対価でも払わざるを得ない。価格は高くなるだろう。すでに、確立された企業には、未来の競争相手を排除する理由がある。
- 「未来の競争相手を排除する」(excluding future competitors)

Patents and copyrights

<< concerning >>

Excluding the future “competitor”

➔ Reducing the possibility of future “innovation”

<< object>>

encourage of disclosure of proprietary intells.

<< implementation >>

protect/regulate for certain period, then go to free/common pool

P2P discussion

- P2P system circulated 5 times (500 %) larger volumes than CDs.
- Reduce of sales volume was 5 %

500% growth vs 5% reduce

1. Creativity and Innovation always builds on the past (創造とイノベーションは常に過去の上に築かれる)
2. The past always tries to control the creativity that builds on it (過去は常にその上に創造されたものと支配しようとする)
3. Free societies enable the future by limiting the past (自由な社会はこの過去の力を制限することで未来を可能にする)
4. Ours is less and less a free society (我々の社会は、日々自由を失っていく)