コンピュータネットワーク
第3回

静岡理工科大学
情報学部　コンピュータシステム学科
幸谷　智紀
https://na-inet.jp/compnet/

本日の概要：通信のお約束事(2/2)

- 符号化
- ハッシュ関数
- 暗号と署名
- SSHによるリモートログイン

符号化：データと自然数(整数)との対応付け＝全単射

- 符号化の一例・・・文字コード
  - 文字集合・・・使用可能な文字の集まり
  - 符号化方法・・・「文字←→bit列」の対応付けの規則

例1)

文字集合→半角英数字(ASCII文字)

符号化方法→7bitで対応

例2)

文字集合→Unicode ・・・ 32bitで全世界の文字を扱う

符号化方法→UTF-8 ・・・ 8bit (1Byte)ごとに区切って表現

ASCII

- 7bitで記号・数字・アルファベットを表わす。

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
	!	″	#	$	%	&	’	(	)	*	+	,	−	.	/	0

49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	@

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P

81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_	`

97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p

113	114	115	116	117	118	119	120	121	122	123	124	125	126
q	r	s	t	u	v	w	x	y	z	{	|	}	~

ハッシュ(Hash)関数：h(x)

- （大きな）データから（一定範囲内の）整数(ハッシュ値)を返す関数

例）MD5→SHA(Secure Hash Algorithm)-1→SHA-256→SHA-3

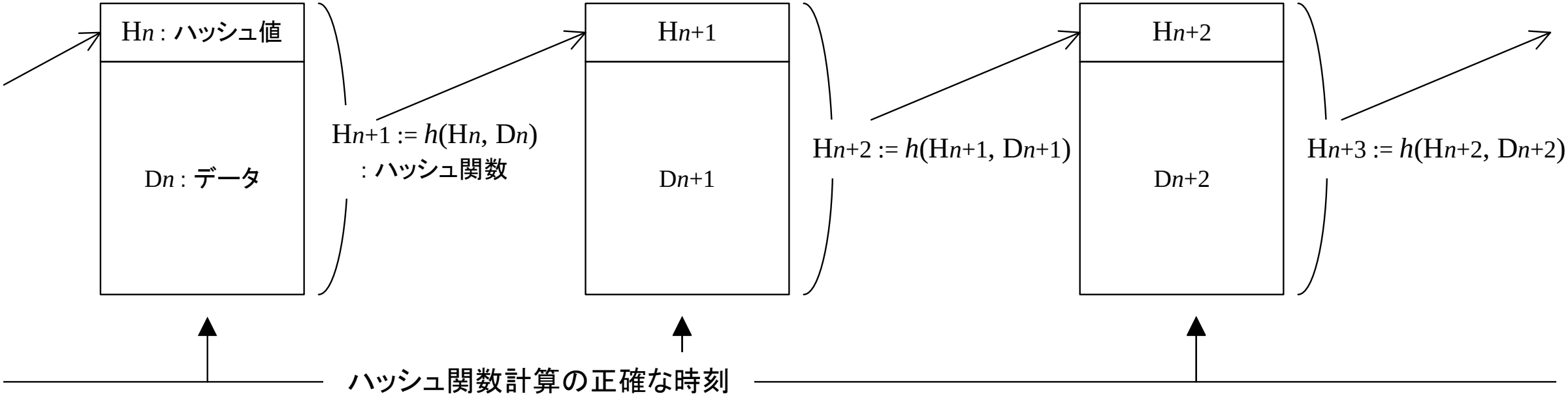
特徴

1. ハッシュ値の出力サイズ（整数の桁数）が同じ→一種の「データ圧縮」
   MD5＝128bit, SHA-1=160bits, SHA-2(SHA-256=256bits, SHA-512=512bit), SHA-3(224～512bit)
2. 一方向性→不可逆データ圧縮
   - ○データ→ハッシュ値
   - ×ハッシュ値→データ（不可逆）
3. 衝突困難性・・・異なるデータから同じハッシュ値が出ない

   N種のデータからM個のデータを取り出し，同じハッシュ値がある確率（M＝SQRT(N)）＝ハッシュ関数のセキュリティ

   MD5:1992年→2006年に衝突　64bitセキュリティ
   SHA-1:1995年→2015年に衝突　80bitセキュリティ
   SHA-256: 2002年→まだ衝突が見つかっていない　128biセキュリティ

ハッシュの応用: ブロックチェーン技術



- Bitcoin(デジタル通貨）を担保するためのシステム
- 正確な時間と共に計算されたハッシュ関数の連なりがblock chainと呼ばれる
- 正確な時間については　https://jjy.nict.go.jp/ntp/　参照。
- 学内では172.16.254.250が時刻合わせサーバ，日本国内では ntp.nict.jp

→「信用」連鎖をハッシュ関数で担保する＝最初が信用できないと・・・？

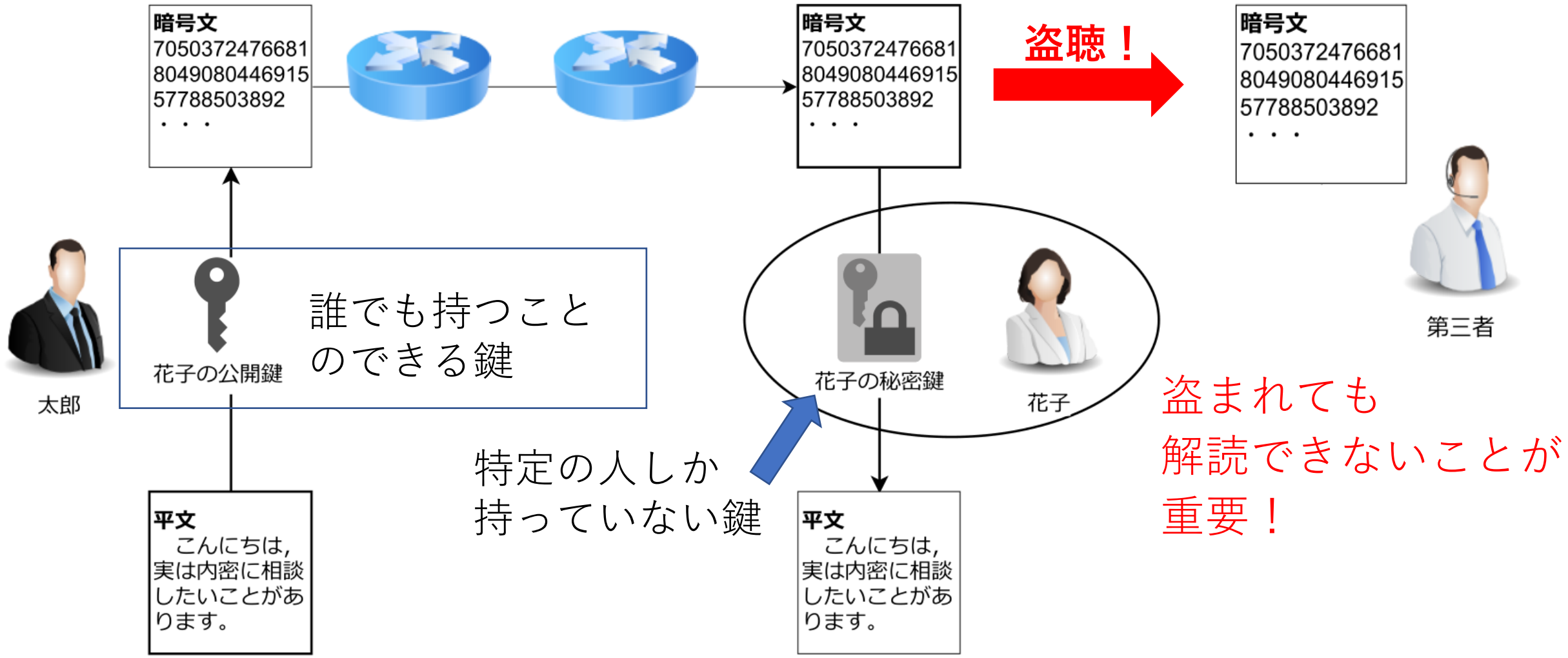
暗号(encryption)

- 平文（plain text, 元のデータ）から判別困難な暗号文(encrypted text)を生成する
  - 暗号化(encrypt)：平文→暗号文
  - 復号化(decrypt)：暗号分→平文

- 符号化されたデータ（整数）のみ扱う
- 「鍵(key)」も整数

- 方式
  - 公開鍵（秘密鍵）暗号：暗号化と復号化で異なる鍵を使用する
  - 共通鍵暗号：お互いに同じ鍵で暗号化・復号化を行う

「暗号化」で守るセキュリティ



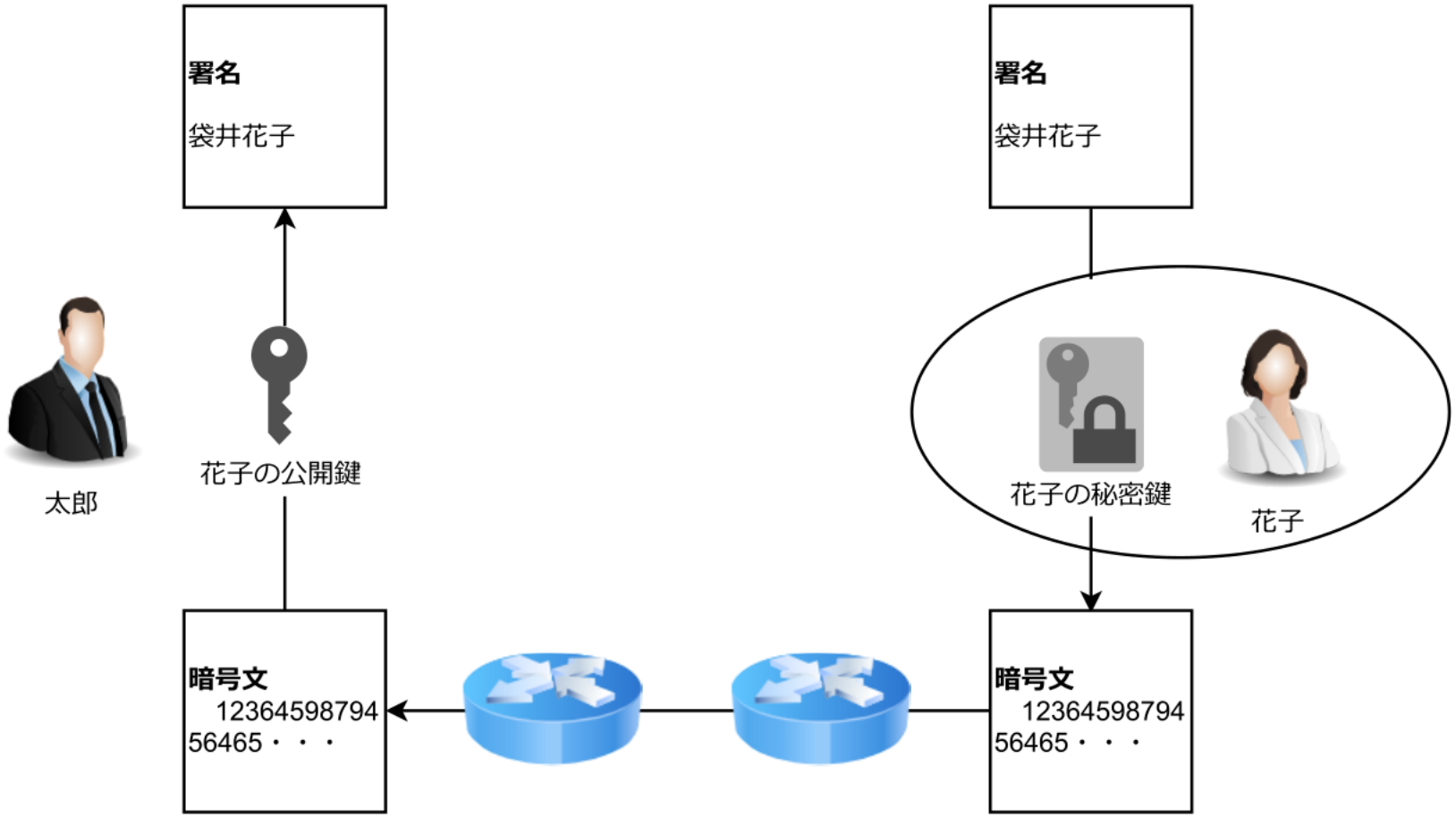
盗まれても
解読できないことが
重要！

「RSA暗号と」は？

RSA暗号は，発案者の頭文字(Rivest-Shamir-Adelman)を取って名付けられた公開鍵暗号方式である。

- 平文：暗号化すべき通常の文字列(plain text)で，整数として表現したものを使用する。
- 公開鍵：下記に述べる手順で予め生成された自然数の組 (e,n) 。所持者に付属するもので，ここでは花子のものとし，誰にでもアクセスできる。
- 秘密鍵：下記に述べる手順で予め訂正された自然数の組(d,n) 。所持者に付属するもので，ここでは花子のものとし，花子以外の人間にはアクセスできないものである。
- 暗号文：平文を暗号化したもの。そのまま平文としては解釈できない整数。

RSA暗号による「署名（サイン）」



公開鍵と秘密鍵：整数の組

① 二つの素数 p, q → $n = pq$ なるべく巨大な数が望ましい

② $l = \varphi(n) = (p-1)(q-1)$ という自然数を作る

③ $\mathrm{GCD}(e, l) = 1$ かつ $1 < e < l$ となるeを決める

→花子の公開鍵(e, n)

④ $ed \bmod l = 1$ かつ $1 < d < l$ となるようdを定める

→花子の秘密鍵(d, n)が生成される

$ed + ly = 1$ となる整数d, yを求めても良い(yは使用しない)。

暗号化 → 暗号文 := 平文$^e \bmod n$

復号化 → 平文 := 暗号文$^d \bmod n$

短い数の例：公開鍵と秘密鍵

① $p = 5, q = 13 \quad \rightarrow \quad n = pq = 65$

② $l = (5-1)(13-1) = 48 = 2^4 \cdot 3$

③例えば$e = 11$とすれば$\mathrm{GCD}(11, 48) = 1$とできる
→花子の公開鍵：$(e, n) = (11, 65)$

④$11d \bmod 48 = 1$となるものとして，例えば$d = 35$
→花子の秘密鍵は$(d, n) = (35, 65)$となる。

短い数の例：暗号化と復号化

花子の公開鍵：$(e, n) = (11, 65)$

花子の秘密鍵：$(d, n) = (35, 65)$

平文＝$55\ (< n)$とする

暗号化→　$55^e \bmod n = 55^{11} \bmod 65 = 35$

13931233916552734375（20桁）

復号化→　$35^d \bmod n = 35^{35} \bmod 65 = 55$ ＝平文

11025074993541486959517864334 1・・・(54桁)

標準(int, long)より長い整数演算→多倍長整数演算

短い数の例：署名

花子の公開鍵：$(e, n) = (11, 65)$

花子の秘密鍵：$(d, n) = (35, 65)$

署名(平文)＝$20\ (< n)$とする

暗号化→　$20^d \bmod n = 20^{35} \bmod 65 = 15$

3435973836・・・　(45桁)

復号化→　$15^e \bmod n = 15^{11} \bmod 65 = 20$ ＝署名

8649755859375 (13桁)

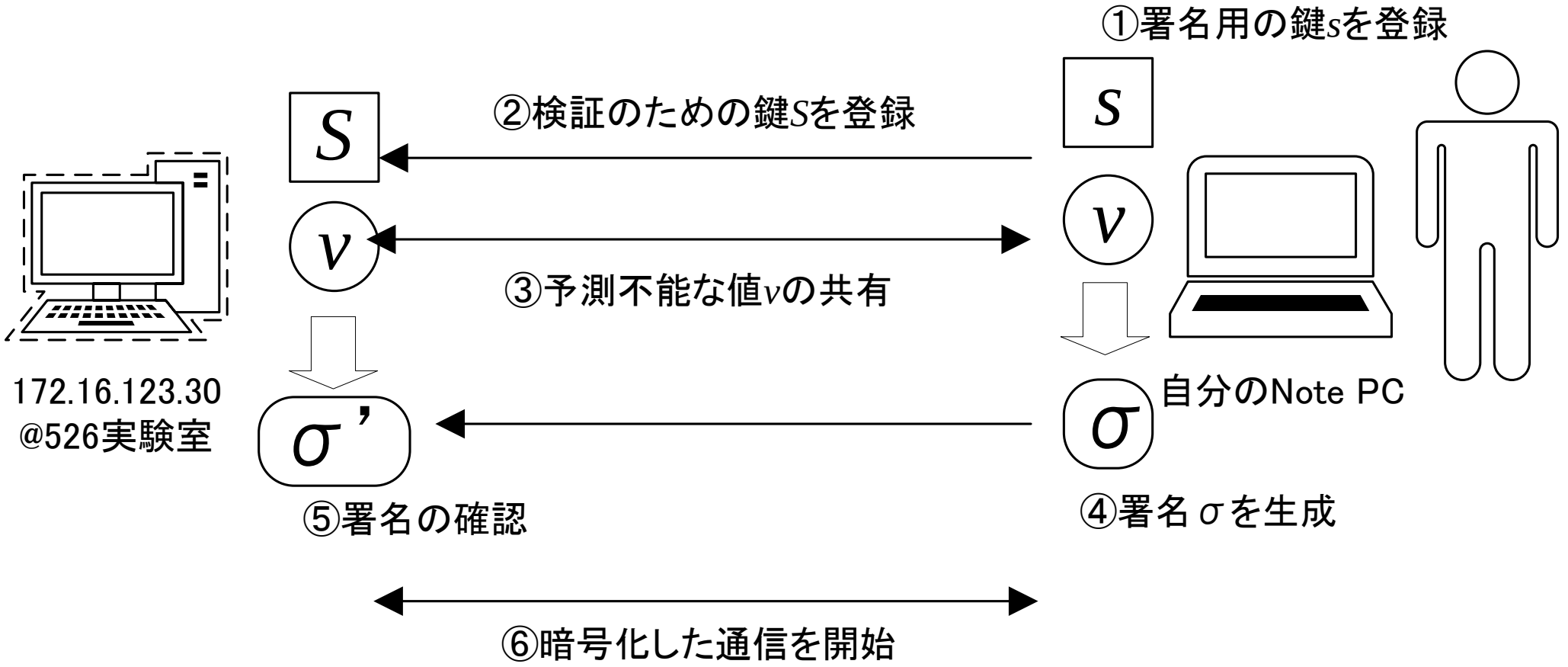
標準(int, long)より長い整数演算→多倍長整数演算

RSA秘密鍵・共通鍵を作る

- Ssh-keygenコマンド: Windowsでも使用可能→SSHログイン時に使用可能 (OpenSSH)

```
tkouya@cs-room526-ryzen37:~/compnet$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tkouya/.ssh/id_rsa): comp_id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in comp_id_rsa
Your public key has been saved in comp_id_rsa.pub
The key fingerprint is:
SHA256:WvUoq7s2neShIQ2IpO2FTEt1lLnML/XToYI1AEGOTzg tkouya@cs-room526-ryzen37
The key's randomart image is:
+---[RSA 3072]----+
|   ...o++o       |
| .o  .o+  .      |
|o* + oE.o ..     |
|o * o ++...oo    |
| . . o oS+.=..   |
|  . . +o=o= .    |
|     ..B.o .     |
|      +.+        |
|     .++         |
+----[SHA256]-----+
```

Secure SHell(SSH)の通信



- 秘密鍵(署名のための鍵）と共有鍵を用いて暗号化した通信路を形成
- 以降は，Linuxマシン(172.16.123.30)との接続をSSHで行う

SSHによるリモートログイン

- ユーザIDとパスワードを使ってログインする。

Your USER ID : guest001 Your Password : [redacted]

1. コマンドプロンプト，もしくは，PowerShellを開く
2. プロンプトから「ssh ユーザID@172.16.123.30」
3. 初回は暗号化の鍵登録を求められるので素直に応じる

```
PS C:\Users\tkouy> ssh guest000@172.16.123.30
The authenticity of host '172.16.123.30 (172.16.123.30)' can't be established.
ED25519 key fingerprint is SHA256:i0kB6Ud60dcudAFxKw6VOAvUhJquh1zhfcQEDZ4t6YM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.123.30' (ED25519) to the list of known hosts.
guest000@172.16.123.30's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

4. ログイン出来たらホームディレクトリのファイルリストを表示する「ls –l」コマンドを打って動作確認。
終了時は「logout」もしくは「exit」

```
guest000@cs-room526-ryzen37:~$ ls -l
合計 16
-rwxr--r--  1 guest000 guest  382  9月 19 17:28 app_first.js
drwxr-xr-x 10 guest000 guest 4096  9月  8 16:40 old_public_html
drwxr-xr-x  5 guest000 guest 4096  9月  9 13:31 public_html
drwx------  3 guest000 guest 4096  9月 19 17:27 snap
guest000@cs-room526-ryzen37:~$ logout
Connection to 172.16.123.30 closed.
PS C:\Users\tkouy>
```

[復習]本日の概要：通信のお約束事(2/2)

- 符号化
- ハッシュ関数
- 暗号と署名
- SSHによるリモートログイン

本日の課題
https://forms.office.com/r/qTM1ZqcFhv

１．以下の□を埋めよ。

①$p = 7, q = 11 \quad \rightarrow \quad n = pq = 77$

②$l = (7-1)(11-1) = 60 = 2^2 \cdot 3 \cdot 5$

③例えば$e = 7$とすれば$\mathrm{GCD}(7, 60) = 1$とできる

→花子の公開鍵：$(e, n) = (7, 77)$

④$7d \bmod 60 = 1$となるものとして，例えば$d =$ ロ

→花子の秘密鍵は$(d, n) = (\text{ロ}, 77)$となる。

２．もう一度，172.16.123.30にリモートログインし，「ls /」コマンドを打ってルートディレクトリのファイルリストを表示し，ログアウトせよ。この過程をスナップショットに撮り，画像ファイルをアップロードせよ。