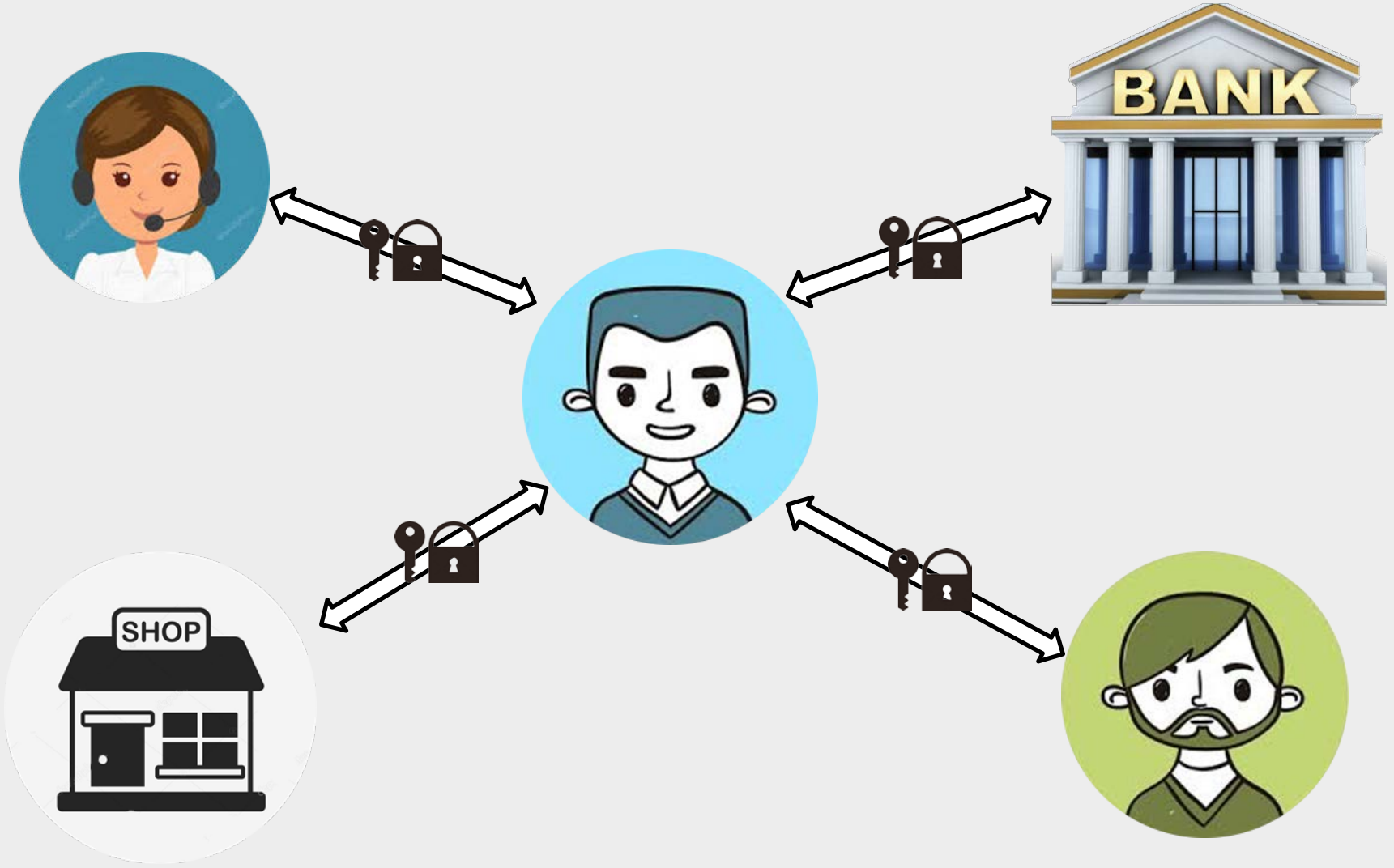
Formalna Weryfikacja Implementacji Algorytmów Kryptograficznych

Tomasz Cabała
Władysław Dudzic

Kryptografia



Systemy kryptograficzne

Model matematyczny → Implementacja

RSA

$$d \equiv e^{-1} \pmod{\varphi(n)}$$

$$ed \equiv 1 \pmod{\varphi(n)}$$

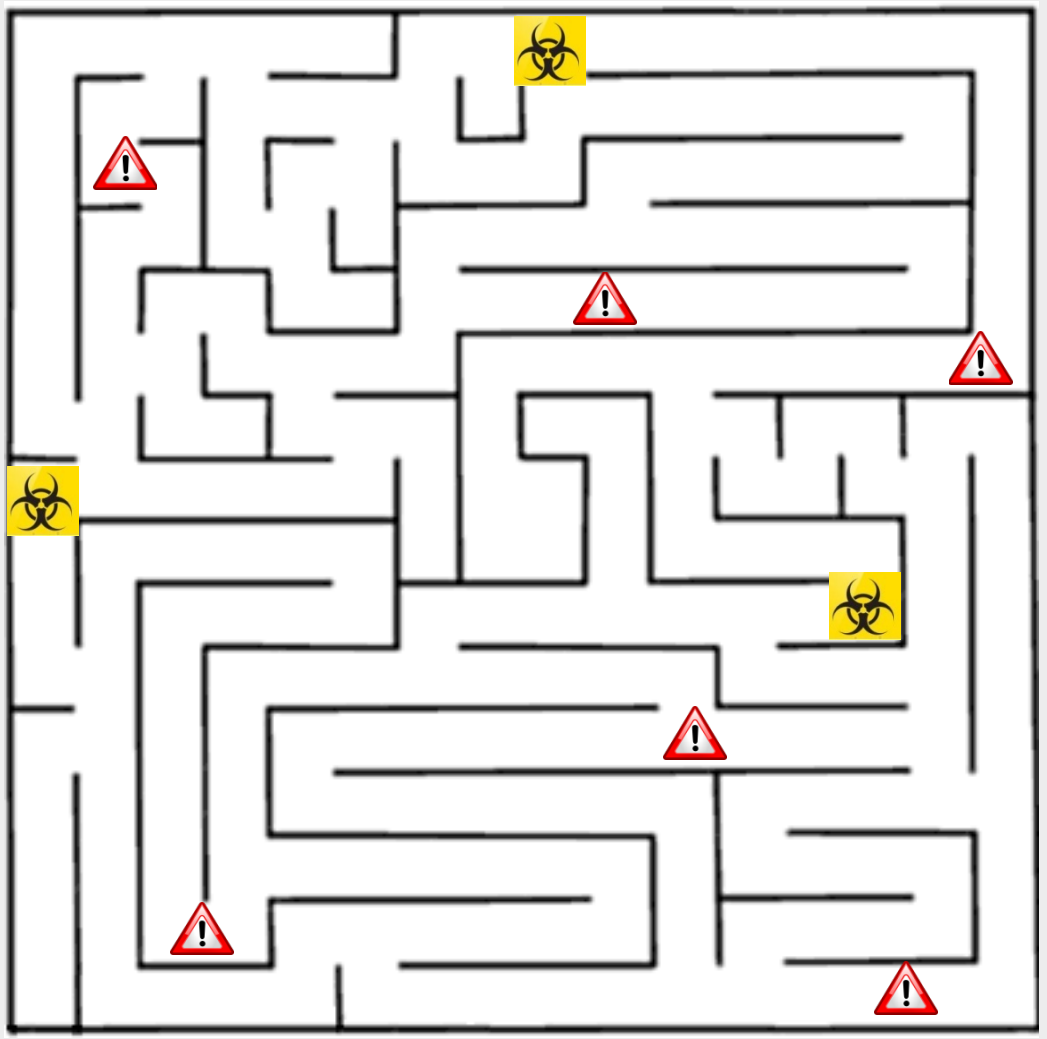
$$ed = k\varphi(n) + 1$$

```
function enEdition(){
    /* Ne rien faire mode edit + preload */
    if( encodeURIComponent(document.location).search(/%26preload%3D/) != -1 )
turn;
    // /&preload=/

    if ( !wgPageName.match(/Discussion.*\/Traduction/) ) return;
    var diff = new Array();
    var status; var pecTraduction; var pecRelecture;
    var avancementTraduction; var avancementRelecture;

/* ************ Parser ************ */
    var params = document.location.search.substr(1, document.location.search.
gth).split('&');
    var i = 0;
    var tmp; var name;
    while ( i < params.length )
    {
        tmp = params[i].split('=');
        name = tmp[0];
        switch( name ) {
            case 'status':
            status = tmp[1];
            break;
            case 'pecTraduction':
```

Sposoby na testowanie oprogramowania dzisiaj



Szkody jakie może wyrządzić oprogramowanie z błędami?

Czym jest weryfikacja formalna ?

Model matematyczny

$$d \equiv e^{-1} \, (mod \, \varphi(n))$$
$$ed \equiv 1 \, (mod \, \varphi(n))$$
$$ed = k\varphi(n) + 1$$

Implementacja

```
function enEdition(){
    /* Ne rien faire mode edit + preload */
    if( encodeURIComponent(document.location).search(/%26preload%3D/) != -1 )
turn;
    // /&preload=/

    if ( !wgPageName.match(/Discussion.*\/Traduction/) ) return;
    var diff = new Array();
    var status; var pecTraduction; var pecRelecture;
    var avancementTraduction; var avancementRelecture;

/* ************ Parser ************ */
    var params = document.location.search.substr(1, document.location.search.
gth).split('&');
    var i = 0;
    var tmp; var name;
    while ( i < params.length )
    {
        tmp = params[i].split('=');
        name = tmp[0];
        switch( name ) {
            case 'status':
            status = tmp[1];
            break;
            case 'pecTraduction':
```

Weryfikacja formalna pozwala na odnalezienie błędów w implementacji

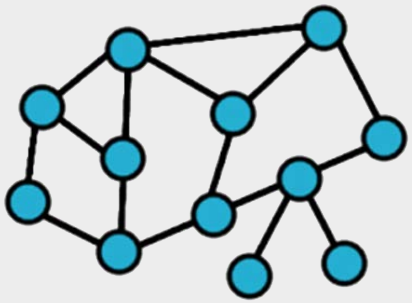
Model matematyczny

RSA

$$d \equiv e^{-1} \ (mod \ \varphi(n))$$

$$ed \equiv 1 \ (mod \ \varphi(n))$$

$$ed = k\varphi(n) + 1$$



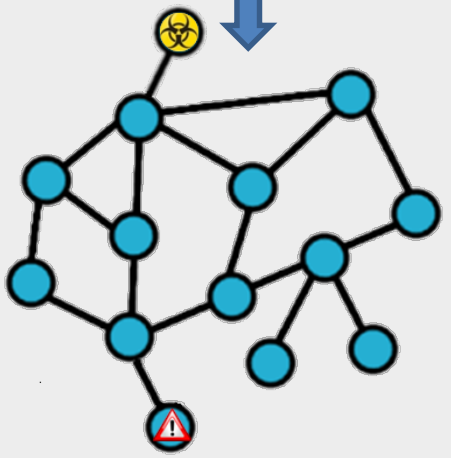
≠

Implementacja

```
function enEdition(){
    /* Ne rien faire mode edit + preload */
    if( encodeURIComponent(document.location).search(/%26preload%3D/) != -1 )
turn;
    // /&preload=/

    if ( !wgPageName.match(/Discussion.*\/Traduction/) ) return;
    var diff = new Array();
    var status; var pecTraduction; var pecRelecture;
    var avancementTraduction; var avancementRelecture;

/* ************ Parser ************ */
    var params = document.location.search.substr(1, document.location.search.
gth).split('&');
    var i = 0;
    var tmp; var name;
    while ( i < params.length )
    {
        tmp = params[i].split('=');
        name = tmp[0];
        switch( name ) {
            case 'status':
            status = tmp[1];
            break;
            case 'pecTraduction':
```



Model matematyczny i poprawna implementacja są równoważne

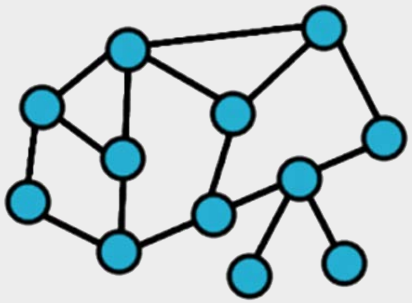
Model matematyczny

RSA

$$d \equiv e^{-1} \ (mod \ \varphi(n))$$

$$ed \equiv 1 \ (mod \ \varphi(n))$$

$$ed = k\varphi(n) + 1$$



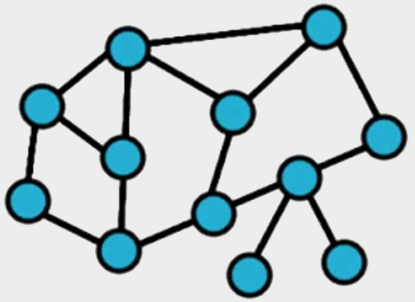
≃

Implementacja

```
function enEdition(){
    /* Ne rien faire mode edit + preload */
    if( encodeURIComponent(document.location).search(/%26preload%3D/) != -1 )
turn;
    // /&preload=/

    if ( !wgPageName.match(/Discussion.*\/Traduction/) ) return;
    var diff = new Array();
    var status; var pecTraduction; var pecRelecture;
    var avancementTraduction; var avancementRelecture;

/* ************ Parser ************ */
    var params = document.location.search.substr(1, document.location.search.
gth).split('&');
    var i = 0;
    var tmp; var name;
    while ( i < params.length )
    {
        tmp = params[i].split('=');
        name = tmp[0];
        switch( name ) {
            case 'status':
            status = tmp[1];
            break;
            case 'pecTraduction':
```



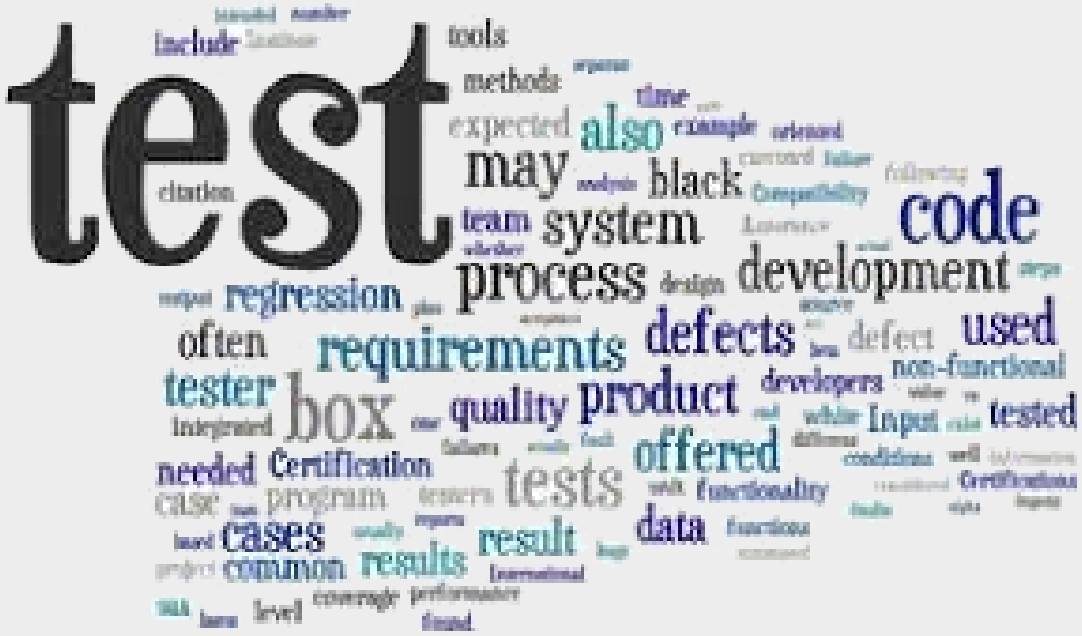
Weryfikacja formalna przykładowych algorytmów kryptograficznych

Algorytm kryptograficzny	Orientacyjny czas weryfikacji [s]
ChaCha-20	75
AES 128/256	7700
ED25519	2118
SHA-512	1565

Cryptol
The Language of Cryptography

Java

C Language



Dziękujemy za uwagę