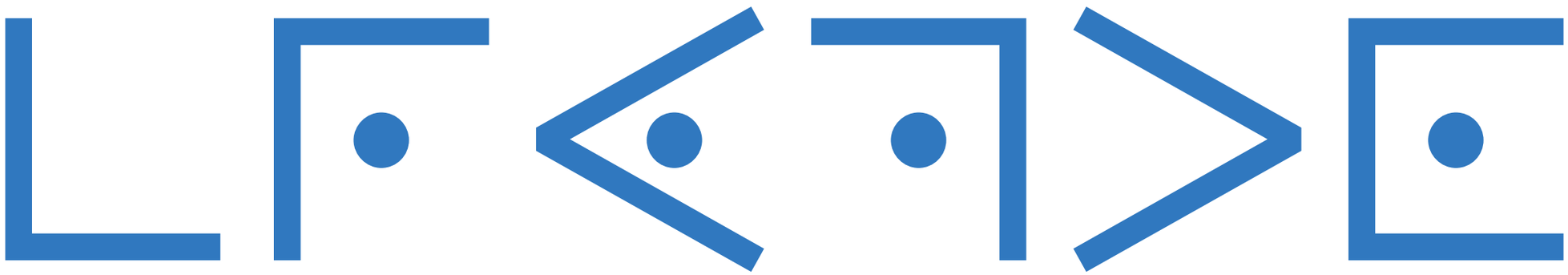


cryptographie à clé publique II

Nicolas Ollinger

M1 informatique — 2024/2025

Formellement

Un schéma de chiffrement à clé publique est défini par 3 algorithmes PPT :

$\text{Gen}(1^k) = (pk, sk)$ algo probabiliste de génération de clés

$\text{Enc}_{pk}(m) = c$ algo probabiliste de chiffrement

$\text{Dec}_{sk}(c) = m$ algo déterministe de déchiffrement

$$\forall (pk, sk) = Gen(1^n) \quad \forall m \quad \text{Dec}_{sk}(\text{Enc}_{pk}(m)) = m$$

Sécurité prouvée

Partir d'un problème algorithmique Π **réputé difficile** (c'est une hypothèse de travail).

Par réduction, montrer que si un adversaire PPT casse le chiffre alors cet adversaire sait aussi résoudre Π en temps polynomial.

Chiffrement ElGamal

ElGamal

Données du schéma : n premier et g non nul, ils peuvent être publiquement connus

Alice choisit une clé secrète s et calcule sa clé publique $y=g^s \pmod{n}$.

Pour chiffrer un message $2 \leq m \leq n-1$, Bob choisit aléatoirement k et et transmet la paire

$$c_1 = g^k \pmod{n} \qquad c_2 = my^k \pmod{n}$$

Alice déchiffre le message en calculant

$$\left(c_1^s\right)^{-1} c_2 = \left(g^{sk}\right)^{-1} my^k = m \pmod{n}$$

Illustration

$n = 467$

$g = 2$

$s = 153$

$m = 331$

$k = 197$

Calculer y, c_1, c_2 pour ensuite retrouver m !

Sécurité prouvée

Pause exercice Démontrer que savoir déchiffrer efficacement ElGamal est équivalent à résoudre efficacement **CDH**.

On pourra pour cela considérer l’existence d’un oracle qui résout un problème et l’utiliser pour résoudre l’autre.

En pratique

La cryptographie à clé publique est généralement bien plus lente à chiffrer que la cryptographie symétrique.

On utilise alors des systèmes hybrides : une clé de session est chiffrée avec une primitive à clé publique, qui permet de déchiffrer les données chiffrées avec une primitive symétrique (avec contrôle d’intégrité !)

Informations sur la page - https://www.root-me.org/

Général | Médias | Permissions | Sécurité

Identité du site web

Site web : www.root-me.org

Propriétaire : Ce site web ne fournit pas d'informations sur son propriétaire.

Vérifiée par : Let's Encrypt

Afficher le certificat

Expire le : 16 février 2021

Détails techniques

Connexion chiffrée (clés TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bits, TLS 1.2)

La page actuellement affichée a été chiffrée avant d'avoir été envoyée sur Internet.

Le chiffrement rend très difficile aux personnes non autorisées la visualisation de la page durant son transit entre ordinateurs. Il est donc très improbable que quelqu'un puisse lire cette page durant son transit sur le réseau.

?

Schéma de chiffrement RSA

Petit théorème de Fermat

Pour tout entier *p* premier, tout entier *m* vérifie :

$$m^p = m \pmod{p}$$

De plus si *p* est premier avec *m* :

$$m^{p-1} = 1 \pmod{p}$$

Théorème d'Euler

L'indicatrice d'Euler $\varphi(n)$ est le nombre d'entiers de 1 à n premiers avec n.

$$\varphi(p) = p - 1 \text{ si } p \text{ est premier}$$

$$\varphi\left(\prod_i p_i^{\alpha_i}\right) = \prod_i \left(p^{\alpha_i} - p^{\alpha_i - 1}\right)$$

$$\varphi(pq) = (p-1)(q-1) \text{ si } p, q \text{ premiers}$$

Théorème de Fermat-Euler

$$a^{\varphi(n)} = 1 \pmod{n} \quad \text{si} \quad \text{pgcd}(a, n) = 1$$

Théorème des restes chinois

Soient n_1, n_2, …, n_k des entiers deux à deux premiers entre eux et soit n leur produit.

Pour tous entiers a_1, a_2, …, a_k, il existe un unique entier x modulo n tel que

$$\begin{aligned} x &= a_1 \pmod{n_1} \\ x &= a_2 \pmod{n_2} \\ &\vdots \\ x &= a_k \pmod{n_k} \end{aligned}$$

Programming Techniques
S.L. Graham, R.L. Rivest*
Editors

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R. L. Rivest, A. Shamir, and L. Adleman
MIT Laboratory for Computer Science
and Department of Mathematics

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:
(1) Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.
(2) A message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds

I. Introduction

The era of "electronic mail" [10] may soon be upon us; we must ensure that two important properties of the current "paper mail" system are preserved: (a) messages are *private*, and (b) messages can be *signed*. We demonstrate in this paper how to build these capabilities into an electronic mail system.

At the heart of our proposal is a new encryption method. This method provides an implementation of a "public-key cryptosystem", an elegant concept invented by Diffie and Hellman [1]. Their article motivated our research, since they presented the concept but not any practical implementation of such a system. Readers familiar with [1] may wish to skip directly to Section V for a description of our method.

II. Public-Key Cryptosystems

In a "public-key cryptosystem" each user places in a public file an encryption procedure E. That is, the public file is a directory giving the encryption procedure of each user. The user keeps secret the details of his corresponding decryption procedure D. These procedures have the following four properties:

(a) Deciphering the enciphered form of a message M yields M. Formally,

Chiffrement RSA

Alice choisit secrètement deux grands nombres premiers $p \neq q$ et calcule $n=pq$. Alice choisit $1<e<\varphi(n)$ premier avec $\varphi(n)=(p-1)(q-1)$ et calcule l'inverse d de e modulo $\varphi(n)$.

Alice publie sa clé publique (n,e) et conserve sa clé privée (n,d).

Pour chiffrer un message $0 \leq m < n$ Bob calcule

$$c = m^e \pmod{n}$$

Pour déchiffrer, Alice calcule

$$c^d = m^{ed} = m \pmod{n}$$

Démonstration

$$de = 1 \pmod{\varphi(n)} \qquad \text{par construction}$$

$$m^{de} = m^{k\varphi(n)+1} \pmod{n} \qquad \text{pour un certain } k$$

$$m^{k\varphi(n)+1} = m \pmod{p} \qquad \text{par Euler ou sinon par } m = 0 \pmod{p}$$

$$m^{k\varphi(n)+1} = m \pmod{q} \qquad \text{par Euler ou sinon par } m = 0 \pmod{q}$$

$$m^{k\varphi(n)+1} = m \pmod{n} \qquad \text{par le théorème des restes chinois}$$

$$c^d = m \pmod{n}$$

Illustration

La clé publique RSA d'Alice est (77,13).

Calculez sa clé privée !

Déchiffrez le message 8.

On pourra s'aider de cette suite de carrés successifs modulo 77 : 8, 64, 15, 71, 36, 64, …

Schéma RSA dit «Textbook»

Gen : sur l'entrée 1^n, générer deux nombres premiers de n bits p et q et deux paramètres d et e pour obtenir les clés (pq,e) et (pq,d).

Enc : étant donnés (N,e) et m calcule

$$c = m^e \pmod{N}$$

Dec : étant donnés (N,d) et c calcule

$$m = c^d \pmod{N}$$

Sécurité de RSA Textbook

Cette façon d'utiliser RSA n'est pas sûre !

…et encore moins si on l'utilise «en mode ECB»

Les problèmes sont nombreux :

- malléabilité : $c_1 c_2 = \mathrm{Enc}_{sk}(m_1 m_2)$
- petits messages et petits exposants
- chiffrement multiple d'un même message

Petit exposant

Pause exercice

Gondolphe a transmis un même message m de 4096 bits à Alice, Bob et Charlie de clés publiques RSA respectives (a,3), (b,3) et (c,3).

Expliquez comment Eve, qui a intercepté les trois messages chiffrés c_1, c_2 et c_3, peut déchiffrer rapidement le message m !

Module commun

Pause exercice

Cunégonde a transmis un même message m de 4096 bits à Alice, Bob de clés publiques RSA respectives (n,a) et (n,b) avec a et b premiers entre eux.

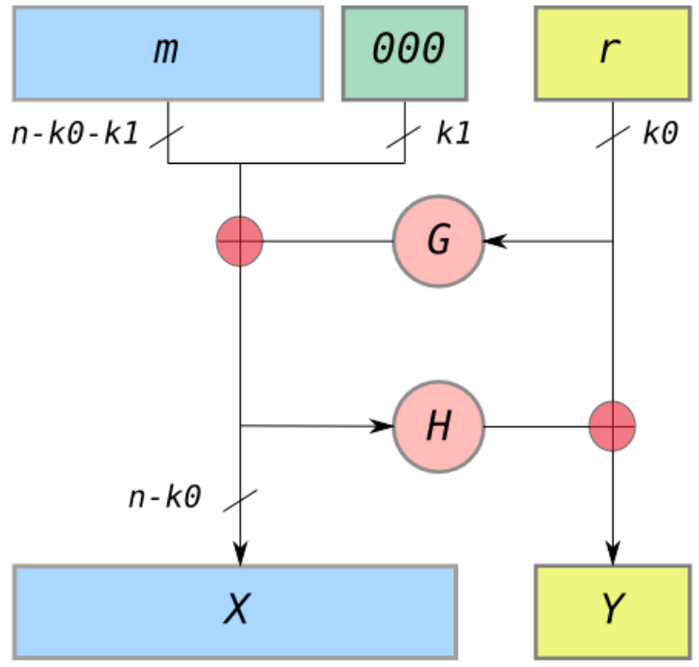
Expliquez comment Eve, qui a intercepté les deux messages chiffrés c_1 et c_2, peut déchiffrer rapidement le message m !

RSA-OAEP

Optimal asymmetric encryption padding

Padding aléatoire par un chiffrement de Feistel.
Remplace le message *m* par un représentant avant chiffrement.

$$m' = m \| 0^k$$
$$X = m' \oplus G(r)$$
$$Y = r \oplus H(X)$$
$$\mathrm{OAEP}(m) = X \| Y$$



RSA-OAEP est sémantiquement sûr sous l'hypothèse RSA.

RSA–OAEP Is Secure under the RSA Assumption*

Eiichiro Fujisaki[1], Tatsuaki Okamoto[1], David Pointcheval[2], and Jacques Stern[2]

[1] NTT Labs, 1-1 Hikarino-oka, Yokosuka-shi, 239-0847 Japan
{fujisaki,okamoto}@isl.ntt.co.jp
[2] Dépt d'Informatique, ENS – CNRS, 45 rue d'Ulm, 75230 Paris Cedex 05, France
{David.Pointcheval,Jacques.Stern}@ens.fr
http://www.di.ens.fr/users/{pointche,stern}

Abstract. Recently Victor Shoup noted that there is a gap in the widely-believed security result of OAEP against adaptive chosen-ciphertext attacks. Moreover, he showed that, presumably, OAEP cannot be proven secure from the *one-wayness* of the underlying trapdoor permutation. This paper establishes another result on the security of OAEP. It proves that OAEP offers semantic security against adaptive chosen-ciphertext attacks, in the random oracle model, under the *partial-domain* one-wayness of the underlying permutation. Therefore, this uses a formally stronger assumption. Nevertheless, since partial-domain one-wayness of the RSA function is equivalent to its (full-domain) one-wayness, it follows that the security of RSA–OAEP can actually be proven under the sole RSA assumption, although the reduction is not tight.

Quelle taille de clé ?

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
(Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	
2016 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

Protection	Symmetric	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
Legacy standard level *Should not be used in new systems*	80	1024	160	1024	160	160
Near term protection *Security for at least ten years (2018-2028)*	128	3072	256	3072	256	256
Long-term protection *Security for thirty to fifty years (2018-2068)*	256	15360	512	15360	512	512

Courbes elliptiques

ECCHacks:
a gentle introduction
to elliptic-curve cryptography

Daniel J. Bernstein
University of Illinois at Chicago &
Technische Universiteit Eindhoven

Tanja Lange
Technische Universiteit Eindhoven

---

ecchacks.cr.yp.to