Publishing Upper Half of RSA Decryption Exponent

Subhamoy Maitra, Santanu Sarkar, and Sourav Sen Gupta

Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, India
{subho,santanu_r}@isical.ac.in, sg.sourav@gmail.com

Abstract. In the perspective of RSA, given small encryption exponent e (e.g., $e = 2^{16} + 1$), the top half of the decryption exponent d can be narrowed down within a small search space. This fact has been previously exploited in RSA cryptanalysis. On the contrary, here we propose certain schemes to exploit this fact towards efficient RSA decryption.

Keywords: Cryptology, Decryption Exponent, Efficient Decryption, Public Key Cryptography, RSA.

1 Introduction

RSA cryptosystem, publicly proposed in 1978 and named after its inventors Ron Rivest, Adi Shamir and Len Adleman, is the most popular Public Key Cryptosystem till date. Let us first briefly describe the RSA scheme [11,13].

Cryptosystem 1 (RSA). *Let us define $N = pq$ where p and q are primes. By definition of the Euler totient function, $\phi(N) = (p-1)(q-1)$.*

- KeyGen: *Choose e co-prime to $\phi(N)$. Find d such that $ed \equiv 1 \bmod \phi(N)$.*
- KeyDist: *Publish public key $\langle N, e\rangle$ and keep private key $\langle N, d\rangle$ secret.*
- Encrypt: *For plaintext $M \in \mathbb{Z}_N$, ciphertext $C = M^e \bmod N$.*
- Decrypt: *For ciphertext C, plaintext $M = C^d \bmod N$.*

The efficiency of encryption and decryption in RSA depends upon the bit-sizes of e and d respectively, and further, both depend on the size of N too, as all the modular operations are done with respect to N. To improve the decryption efficiency of RSA, another variant of RSA was proposed that uses the Chinese Remainder Theorem (CRT). This is the most widely used variant of RSA in practice and is known as CRT-RSA [10,18].

Preliminaries. Before proceeding further, let us go through some preliminary discussions. For notational purpose, we denote the number of bits in an integer i by l_i; i.e., $l_i = \lceil \log_2 i \rceil$ when i is not a power of 2 and $l_i = \log_2 i + 1$, when i is a power of 2. By *Small* e, we mean $e = 2^{16} + 1$ or around that range, which is popularly used for fast RSA encryption.

I. Echizen, N. Kunihiro, and R. Sasaki (Eds.): IWSEC 2010, LNCS 6434, pp. 25–39, 2010.
© Springer-Verlag Berlin Heidelberg 2010

Fact 1. *For* Small e*, the top half of* d *can be estimated efficiently.*

Proof. The RSA equation $ed = k\phi(N) + 1$ translates to $ed = k(N+1) - k(p+q) + 1$, where $l_k \approx l_e$ and $l_d \approx l_N$. In cases where e is *Small*, so is k, and hence can be found using a brute force search. Thus one can estimate d as follows.

$$d = \frac{k}{e}(N+1) + \frac{1}{e} - \frac{k}{e}(p+q) \approx \frac{k}{e}(N+1) + \frac{1}{e}$$

The error in this approximation is $\frac{k}{e}(p+q) < p+q$ as $1 < k < e$. Thus, considering that the primes p and q are of same bit-size, the error is $O(\sqrt{N})$, that is one gets an approximation with error-size less than or equal to $\max(l_p, l_q) \approx \frac{1}{2}l_N \approx \frac{1}{2}l_d$. If we write $d = d_0 + d_1$ with $d_0 = \lceil \frac{k}{e}(N+1) + \frac{1}{e} \rceil$, then $|d - d_0| < 2^{l_N/2}$, which implies that d_0 estimates the top half of d correctly and $d_1 \equiv d \pmod{2^{l_N/2}}$. Thus, for various values of k in the range $1 \leq k < e$, we get those many possibilities for the upper half of d, allowing for an efficient estimate. □

Our Motivation. The estimation of d stated in Fact 1 has been exploited in literature to propose partial key exposure attacks on RSA [2]. Our motivation though is to use this estimation in a constructive way. As one can estimate the half of the bits of d in most significant side anyway in cases where e is *Small*, there is no harm in choosing that top half on our own and to make it public. A few interesting questions come up in this direction.

- Can one choose the most significant half of the bits of d to make RSA decryption more efficient than in the case for general RSA?
- Can one choose the most significant half of the bits of d to *personalize* RSA in some way?
- Can one choose the least significant half of the bits of d in some way (no constraint on the most significant half) so that higher workload can be transferred to a server in case of a server-aided decryption?

Our Contribution. In this paper, we shall answer these questions one by one. First, in Section 2, we propose a scheme for RSA where one can choose around half of the bits of d in most significant side on his/her own, simply to make RSA decryption faster for *Small* e. It is important to note that our result does not compete with fast CRT-RSA decryption; this is only to highlight how simply the general RSA decryption can be made more efficient through this idea. Next, in Section 3, we answer the second question by proposing a *personalized* model of RSA by letting the user choose the most significant half of d on his own. We provide an answer to the third question in Section 4 and illustrate one of its potential applications in the form of a new RSA scheme for low end devices.

In modern cryptography, the issue of cryptographic implementation on low end devices is an emerging field of research. The importance of efficiency comes from the computational constraints in low end hand-held devices like smart cards, phones or PDAs. One knows that sometimes the low end devices (M, say)