International Journal of Internet, Broadcasting and Communication Vol.10 No.3 11-25 (2018)
http://dx.doi.org/10.7236/IJIBC.2018.10.3.11

IJIBC 18-3-2

On Recovering Erased RSA Private Key Bits

Yoo-Jin Baek

Department of Information Security, Woosuk University, Korea
yoojin.baek@gmail.com

Abstract

While being believed that decrypting any RSA ciphertext is as hard as factorizing the RSA modulus, it was also shown that, if additional information is available, breaking the RSA cryptosystem may be much easier than factoring. For example, Coppersmith showed that, given the 1/2 fraction of the least or the most significant bits of one of two RSA primes, one can factorize the RSA modulus very efficiently, using the lattice-based technique. More recently, introducing the so called cold boot attack, Halderman et al. showed that one can recover cryptographic keys from a decayed DRAM image. And, following up this result, Heninger and Shacham presented a polynomial-time attack which, given 0.27-fraction of the RSA private key of the form (p, q, d, d_p, d_q)*, can recover the whole key, provided that the given bits are uniformly distributed. And, based on the work of Heninger and Shacham, this paper presents a different approach for recovering RSA private key bits from decayed key information, under the assumption that some random portion of the private key bits is known. More precisely, we present the algorithm of recovering RSA private key bits from erased key material and elaborate the formula of describing the number of partially-recovered RSA private key candidates in terms of the given erasure rate. Then, the result is justified by some extensive experiments.*

Keywords: *RSA, Side-Channel Attack, Cold-Boot Attack, Key Recovery*

1. Introduction

RSA is the most widely used public key cryptosystem and makes use of the property that, given two distinct primes p, q and two exponents e, d satisfying $ed = 1 \bmod (p-1)(q-1)$, $(m^e)^d = m \bmod pq$ for any integer m ([9]). Letting $N = pq$, (N, e) is then called an RSA public key and d is called an RSA private key. While only d can be used as an RSA private key, it is also recommended to use the tuple $(p, q, d, d_p, d_q, q^{-1} \bmod p)$ as a private key for the fast RSA decryption, where $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$ ([11]). And, since the component $q^{-1} \bmod p$ is not required in the subsequent analysis, we will refer the tuple (p, q, d, d_p, d_q) as an RSA private key for convenience.

While it is believed that, given a ciphertext c, finding the plaintext m satisfying $c = m^e \bmod N$ is as hard as factorizing N, it was also shown that, if additional information is available, breaking the RSA cryptosystem may be easier than factorizing. For example, Coppersmith showed that, given the $1/2$ fraction

Manuscript Received: May. 18, 2018 / Revised: May. 26, 2018 / Accepted: June. 1, 2018
Corresponding Author: yoojin.baek@gmail.com
Tel: +82-63-290-1221, Fax: +82-63-290-1518
Department of Information Security, Woosuk University, Korea

of the least or the most significant bits of one of two RSA primes, then one can factorize the RSA modulus, using the lattice-based technique ([2]). More recently, introducing the so called cold boot attack, Halderman et al. showed that one can extract some information from powered-off DRAM and then recover cryptographic keys from the information, if any. And, following up this result, Heninger and Shacham presented a polynomial-time attack which, given a 0.27-fraction of the RSA private key of the form (p, q, d, d_p, d_q), a 0.42-fraction of (p, q, d) or a 0.57-fraction of (p, q), can reconstruct the whole RSA private key, provided that the given bits are uniformly distributed over the private key ([5]). It is notable that the technique in [5] is different from that of [2] in that, while the Coppersmith's method requires the attacker to know consecutive least or most significant bits, the method of Heninger and Shacham does not assume to have any knowledge for the bit positions in which the attacker knows the exact bit values. In Crypto 2010, Henecka et al. also studied the RSA key recovery attack, under which each bit of the RSA private key was assumed to be flipped with some probability ([4]). And, they could correct the errors whenever the error rate is less than 0.237 in case of (p, q, d, d_p, d_q), 0.16 in (p, q, d) or 0.084 in (p, q).

Based on the work of [5], this paper presents a different approach for recovering the RSA private key bits from the decayed key image, under the assumption that some random portion of the private key bits is known. More precisely, we present an algorithm of recovering RSA private key bits from the erased key material and elaborate a general formula of describing the number of partially-recovered RSA private key candidates in terms of the given erasure rate. Then, the result is justified with some extensive experiments.

This paper is organized as follows. In Section 2, we give an overview of the side-channel attack and the cold-boot attack and introduce some previous results on the cold-boot attacks. Section 3 is the place in which we clearly present new RSA key recovery analysis and give some experimental results. The conclusion is drawn in Section 4.

2. Preliminaries

2.1 Side-Channel Attack

The side-channel attack makes use of side-channel information to extract some secret material in cryptographic devices. Various side-channel information were used for the attacks, among which there are timing, power consumption, electro-magnetic emanation and (intentional or unintentional) faulty outputs of cryptographic operations. Since first introduced by Kocher et al. ([7], [8]), a wide variety of side-channel attack methods and countermeasures were published in the literature.

Recently, another kind of side-channel attack techniques, the so called cold boot attack, was published, based on the DRAM data-retention property ([3]). That is, contrary to the common belief, DRAM which is used for the main memory of various computing devices retains its data even when it is powered-off. In particular, the data-retaining time can increase if DRAM is cooled down. Before cryptographic operations are performed, the cryptographic keys are usually uploaded into the main memory. Hence, if DRAM is suddenly powered-off, it would hold very sensitive information such as cryptographic keys and the cold boot attack extracts and recovers this information.

2.2 Previous Result

Cryptographic key recovery techniques under the cold boot attack scenario can be classified with various ways. First, in terms of the target algorithms, they can be for secret key algorithms or public key algorithms. Next, when the cryptographic key material is extracted from the powered-off DRAM, the content is not

guaranteed to be as it was, due to some physical properties of DRAM. And, the key recovery algorithms can be categorized in terms of how the bits have been decayed after extraction ([12]): with some probability, some bits can have been erased (erasure model) or can have been flipped (correction model). Also, the correction model can be subdivided into: letting as δ_{1-b} the probability that the bit b is flipped into $1-b$ after extraction, the symmetric correction model assumes $\delta_0 = \delta_1$ while the asymmetric correction model assumes $\delta_0 \neq \delta_1$. And, the asymmetric correction model has two subclasses, the asymetric perfect model assuming that one of δ_0 and δ_1 is zero and the asymmetric imperfect model assuming that both δ_0 and δ_1 are non-zero.

For the symmetric key algorithms, Halderman et al. presented basic key recovery algorithms for DES and AES under the asymmetric imperfect model with $\delta_0 \gg \delta_1 \approx 0$ ([3]). Especially, they showed that the AES 128-bit key can be re-constructed in a second for $\delta_0 = 0.15, \delta_1 = 0.001$ and within 30 seconds for $\delta_0 = 0.3, \delta_1 = 0.001$. Tsow presented an AES key recovery algorithm under the asymmetric perfect model ([12]) and Kamal and Youseff showed that the AES key recovery problem under the asymmetric perfect model can be converted into the Boolean Satisfiability Problem ([6]). Also, Albrecht et al. proposed a new method transforming the key recovery problem into the problem of solving the set of non-linear algebraic equations with noise ([1]).

For the public key algorithms, Halderman et al. also presented basic key recovery algorithms for RSA under the asymmetric perfect model ([3]). And, in Crypto 2009, Heninger and Shacham provided the algorithm which can factorize the RSA modulus N in polynomial time when the attacker is given 0.27 fraction of random bits of (p, q, d, d_p, d_q), 0.42 fraction of random bits of (p, q, d), or 0.57 fraction of random bits of (p, q) ([5]). Also, in Crypto 2010, Henecka et al. studied the RSA key recovery under the correction model in which each bit of the RSA private key is flipped with some probability and proved that one can correct the errors in the RSA private key when the error rate is less than 0.237 when (p, q, d, d_p, d_q) are known with error, less than 0.160 when (p, q, d) are known with error, or less than 0.084 when (p, q) are known with error. And, in [10], Paterson et al. applied the coding-theoretic methodology to the problem of noisy RSA key recovery and derived some theoretical bound on the performance of the previous algorithms.

2.3 Algorithm of Henninger-Shacham

Since the result of this paper is based on the work of [5], this section will review it in more detail.

For the fast RSA decryption, the PKCS \#1 ([11]) recommends to use the private key of the form $(p, q, d, d_p, d_q, q^{-1} \bmod p)$, where

$$\begin{aligned} N &= pq \\ ed &= 1 \; mod \; (p-1)(q-1) \\ ed_p &= 1 \bmod (p-1) \\ ed_q &= 1 \bmod (q-1). \end{aligned} \tag{1}$$

Noting that the knowledge of any of $p, q, d, d_p, d_q, q^{-1} \bmod p$ is sufficient to reveal the factorization of N, Heninger and Shacham then introduced the method which can reconstruct a corrupted RSA private key. To do this, they first assumed that the public exponent e is so small that one can find the exact values of

$k, k_p, k_q \in \boldsymbol{Z}$ satisfying

$$\begin{aligned} ed &= 1 + k(p-1)(q-1) \\ ed_p &= 1 + k_p(p-1) \\ ed_q &= 1 + k_q(q-1). \end{aligned} \tag{2}$$

with $O(e)$ trials, which means that, without loss of generality, k, k_p, k_q are presumably known.

Now, denoting $x[i]$ as the i-th bit of a positive integer x (so, $x[0]$ stands for the the least significant bit of x), it was found that the equations (1) lead to the following modular equations: for $i \geq 1$,

$$\begin{aligned} p[i] + q[i] &= (N - p'q')[i] \bmod 2 \\ d[i] + p[i] + q[i] &= (k(p'-1)(q'-1) + 1 - ed')[i] \bmod 2 \\ d_p[i] + p[i] &= \left(k_p(p'-1) + 1 - ed'_p\right)[i] \bmod 2 \\ d_q[i] + q[i] &= \left(k_q(q'-1) + 1 - ed'_q\right)[i] \bmod 2, \end{aligned} \tag{3}$$

where p', q', d', d'_p, d'_q are the values such that

$$\begin{aligned} N &= p'q' \bmod 2^i \\ ed' &= 1 + k(p'-1)(q'-1) \bmod 2^i \\ ed'_p &= 1 + k_p(p'-1) \bmod 2^i \\ ed'_q &= 1 + k_q(q'-1) \bmod 2^i. \end{aligned} \tag{4}$$

And, using these equations, Heninger and Shacham could recover the bits of p, q, d, d_p, d_q from the least significant bit position upwards. In doing so, they compared the recovered bits with the known key bits, thus finally could show that the number of private key candidates does not increase so rapidly if the erasure rate is not so large.

3. New Analysis

As in [5], this paper assumes that the public exponent e is so small that we can efficiently compute the exact values of k, k_p, k_q satisfying (2).

Given the δ-fraction of (p, q, d, d_p, d_q) for $0 \leq \delta \leq 1$, let W_i be the set of solutions $(x, y, z, u, v) \in \boldsymbol{Z}^5$ such that

$$\begin{aligned} N &= xy \bmod 2^i \\ ez &= 1 + k(x-1)(y-1) \bmod 2^i \\ eu &= 1 + k_p(x-1) \bmod 2^i \\ ev &= 1 + k_q(y-1) \bmod 2^i. \end{aligned}$$

and let S_i be the size of W_i. Clearly, if N is an n-bit integer, (p, q, d, d_p, d_q) should be contained in W_n. And, since all p, q, d, d_p, d_q are odd integers, W_1 is equal to $\{(1,1,1,1,1)\}$. The main goal of this paper is

to estimate S_i in terms of δ.

Suppose that $(x, y, z, u, v) \in W_i$. Then, applying (3), one can show that, for $(x_i, y_i, z_i, u_i, v_i) \in \{0,1\}^5$, $(x_i 2^i + x, y_i 2^i + y, z_i 2^i + z, u_i 2^i + u, v_i 2^i + v)$ is contained in W_{i+1}, if x_i, y_i, z_i, u_i, v_i are satisfying

$$\begin{aligned} x_i + y_i &= (N - xy)[i] \bmod 2 \\ z_i + x_i + y_i &= (k(x-1)(y-1) + 1 - ez)[i] \bmod 2 \\ u_i + x_i &= \left(k_p(x-1) + 1 - eu\right)[i] \bmod 2 \\ v_i + y_i &= \left(k_q(y-1) + 1 - ev\right)[i] \bmod 2 \end{aligned} \tag{5}$$

And, $(x_i 2^i + x, y_i 2^i + y, z_i 2^i + z, u_i 2^i + u, v_i 2^i + v)$ will be called an *extended* solution of $(x, y, z, u, v) \in W_i$ if it satisfies (5).

Note that the system of equations (5) is under-defined, so we can find exactly two solutions $(x_i, y_i, z_i, u_i, v_i$ for given (x, y, z, u, v). Also, even though $(p \bmod 2^i, q \bmod 2^i, d \bmod 2^i, d_p \bmod 2^i, d_q \bmod 2^i)$ is in W_i, there may be other elements of W_i. Thus, we will call $(x, y, z, u, v) \in W_i$ the *good* (partial) solution if it is equal to $(p \bmod 2^i, q \bmod 2^i, d \bmod 2^i, d_p \bmod 2^i, d_q \bmod 2^i)$ and a *bad* solution, otherwise.

For the ease of analysis, we will also need the following conjecture about the behavior of bad solutions, the validity of which was empirically verified in [5].

Conjecture The probability that an extended solution from a bad solution satisfies each equation of (5) is independently 1/2.

Based on all these assumptions and arguments, the following algorithm can work for recovering the original RSA private key from the decayed key information.

Algorithm 1 (Erasure Correction for (p, q, d, d_p, d_q))

Input : (N, e) and δ-fraction of (p, q, d, d_p, d_q)

Output : (p, q, d, d_p, d_q)

1. $W_1 \leftarrow \{(1,1,1,1,1)\}$
2. For $i = 1$ to $n - 1$
   A. For each $(x, y, z, u, v) \in W_i$, find two solutions $(x_i, y_i, z_i, u_i, v_i)$ of (5). And, if all x_i, y_i, z_i, u_i, v_i match with the corresponding known bits of p, q, d, d_p, d_q, then add it to W_{i+1}. Otherwise, discard it.
3. For each $(x, y, z, u, v) \in W_n$, if $xy = N$, then output (x, y, z, u, v).

Remark In Algorithm 1, the inner loop in Step 2 is designed to terminate when $i = n$. But, due to the

result of Coppersmith in [2], the final loop index can be reduced to $n/4$. That is, after completing the loop at $i = n/4$, we can apply the lattice based technique to the x-component of each element in $W_{n/4}$ and then check if the resulting value does divide N.

Cleary, $S_1 = 1$ and we can now estimate S_{i+1} from S_i. To do this, we should consider 32 cases according to whether the bits $p[i], q[i], d[i], d_p[i], d_q[i]$ are known or not.

Case 1 (all $p[i], q[i], d[i], d_p[i], d_q[i]$ are unknown) In this case, we can find exactly 2 solutions $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$. Thus, $S_{i+1} = 2S_i$ with the probability $(1-\delta)^5$.

Case 2 (only $p[i]$ is known) In this case, we can find one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where x_i should be equal to $p[i]$. Thus, $S_{i+1} = S_i$ with the probability $\delta(1-\delta)^4$.

Case 3 (only $q[i]$ is known) In this case, the similar argument as in Case 2 can be applied. Thus, $S_{i+1} = S_i$ with the probability $\delta(1-\delta)^4$.

Case 4 (only $d[i]$ is known) In this case, we first substitute the value of $d[i]$ for z_i in the second equation of (5) and check if the first and second equations of (5) are compatible (which means that both equations give the same solution). Clearly, if (x, y, z) is a *good* solution, then those equations are compatible. Otherwise, from Conjecture above, they are compatible with the probability 1/2. Also, if the equations are compatible, we can find two solutions $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where z_i should be equal to the known $d[i]$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 2 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 2\right) S_i = S_i + 1$ with the probability $\delta(1-\delta)^4$.

Case 5 (only $d_p[i]$ is known) In this case, we can find one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where u_i should be equal to $d_p[i]$. Thus, $S_{i+1} = S_i$ with the probability $\delta(1-\delta)^4$.

Case 6 (only $d_q[i]$ is known)In this case, the similar argument as in Case 5 can be applied. Thus, $S_{i+1} = S_i$ with the probability $\delta(1-\delta)^4$.

Case 7 (only $p[i]$ and $q[i]$ are known) In this case, we first substitute the values of $p[i]$ and $q[i]$ for x_i and y_i in the first equation of (5) and check if the equation holds. Clearly, if (x, y, z, u, v) is a *good* solution, the first equation should hold with these values. Otherwise, from Conjecture above, it holds with the probability 1/2. Also, if the first equation holds, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, y_i) should be equal to $(p[i], q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 8 (only $p[i]$ and $d[i]$ are known) In this case, we first substitute the values of $p[i]$ and $d[i]$ for x_i and z_i in the first and second equations of (5) and check if these equations are compatible. Clearly, if (x, y, z, u, v) is a *good* solution, then these equations are compatible. Otherwise, from Conjecture above, they are compatible with the probability 1/2. Also, if these equations are compatible, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, z_i) should be equal to $(p[i], d[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 9 (only $p[i]$ and $d_p[i]$ are known) In this case, we first substitute the values of $p[i]$ and $d_p[i]$

for x_i and u_i in the third equation of (5) and check if the equation holds. Clearly, if (x, y, z, u, v) is a *good* solution, the third equation should hold with these values. Otherwise, from Conjecture above, it holds with the probability 1/2. Also, if the third equation holds, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, u_i) should be equal to $(p[i], d_p[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 10 (only $p[i]$ and $d_q[i]$ are known) In this case, we first substitute the values of $p[i]$ and $d_q[i]$ for x_i and v_i in the first and fourth equations of (5) and check if these equations are compatible. Clearly, if (x, y, z, u, v) is a *good* solution, then these equations are compatible. Otherwise, from Conjecture above, they are compatible with the probability 1/2. Also, if these equations are compatible, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, v_i) should be equal to $(p[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 11 (only $q[i]$ and $d[i]$ are known) In this case, the similar argument as in Case 8 can be applied. Thus, $S_{i+1} = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 12 (only $q[i]$ and $d_p[i]$ are known) In this case, the similar argument as in Case 10 can be applied. Thus, $S_{i+1} = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 13 (only $q[i]$ and $d_q[i]$ are known) In this case, the similar argument as in Case 9 can be applied. Thus, $S_{i+1} = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 14 (only $d[i]$ and $d_p[i]$ are known) In this case, we first substitute the value of $d[i]$ and $d_p[i]$ for z_i and u_i in the second and the third equations of (5) and check if the first, the second and the third equations of (5) are compatible. Clearly, if (x, y, z, u, v) is a *good* solution, then these equations are compatible. Otherwise, from Conjecture above, they are compatible with the probability 1/2. Also, if these equations are compatible, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (z_i, u_i) should be equal to $(d[i], d_p[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 15 (only $d[i]$ and $d_q[i]$ are known) In this case, the similar argument as in Case 14 can be applied. Thus, $S_{i+1} = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 16 (only $d_p[i]$ and $d_q[i]$ are known) In this case, we first substitute the values of $d_p[i]$ and $d_q[i]$ for u_i and v_i in the third and fourth equations of (5) and check if the first, the third and the fourth equations are compatible. Clearly, if (x, y, z, u, v) is a *good* solution, then these equations are compatible. Otherwise, from Conjecture above, they are compatible with the probability 1/2. Also, if these equations are compatible, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (u_i, v_i) should be equal to $(d_p[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1-\delta)^3$.

Case 17 (only $p[i], q[i], d[i]$ are known) In this case, we first substitute the values of $p[i], q[i]$ and $d[i]$ for x_i, y_i and z_i in the first and second equations of (5) and check if these equations holds. Clearly, if

(x, y, z, u, v) is a *good* solution, these equations should hold with these values. Otherwise, from Conjecture above, they hold with the probability 1/4. Also, if these equations hold, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, y_i, z_i) should be equal to $(p[i], q[i], d[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{4} \cdot 1\right) S_i = \frac{S_i + 3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 18 (only $p[i], q[i], d_p[i]$ are known) In this case, we first substitute the values of $p[i], q[i]$ and $d_p[i]$ for x_i, y_i and u_i in the first and the third equations of (5) and check if these equations holds. Clearly, if (x, y, z, u, v) is a *good* solution, these equations should hold with these values. Otherwise, from Conjecture above, it holds with the probability 1/4. Also, if these equations hold, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, y_i, u_i) should be equal to $(p[i], q[i], d_p[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{4} \cdot 1\right) S_i = \frac{S_i + 3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 19 (only $p[i], q[i], d_q[i]$ are known) In this case, the similar argument as in Case 18 can be applied. Thus, $S_{i+1} = \frac{S_i + 3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 20 (only $p[i], d[i], d_p[i]$ are known) In this case, we first substitute the values of $p[i], d[i]$ and $d_p[i]$ for x_i, z_i and u_i in the first, the second and the third equations of (5) and check if the first and the second equations are compatible and the third equation holds. Clearly, if (x, y, z, u, v) is a *good* solution, the first and second equations are compatible and the third equation holds with these values. Otherwise, from Conjecture above, the first and second equations are compatible and the third equation holds with the probability 1/4. Also, if the first and the second equations are compatible and the third equation holds, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, z_i, u_i) should be equal to $(p[i], d[i], d_p[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{4} \cdot 1\right) S_i = \frac{S_i + 3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 21 (only $p[i], d[i], d_q[i]$ are known) In this case, we first substitute the values of $p[i], d[i]$ and $d_q[i]$ for x_i, z_i and v_i in the first, the second and the fourth equations of (5) and check if these equations are compatible. Clearly, if (x, y, z, u, v) is a *good* solution, then these equations are compatible. Otherwise, from Conjecture above, they are compatible with the probability 1/4. Also, if these equations are compatible, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, z_i, v_i) should be equal to $(p[i], d[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{4} \cdot 1\right) S_i = \frac{S_i + 3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 22 (only $p[i], d_p[i], d_q[i]$ are known) In this case, we first substitute the values of $p[i], d_p[i]$ and $d_q[i]$ for x_i, u_i and v_i in the first, the third and the fourth equations of (5) and check if the first and the fourth equations are compatible and the third equation holds. Clearly, if (x, y, z, u, v) is a *good* solution, the first and the fourth equations are compatible and the third equation holds with these values. Otherwise, from Conjecture above, the first and the fourth equations are compatible and the third equation holds with the probability 1/4. Also, if the first and the fourth equations are compatible and the third equation holds, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, u_i, v_i) should be equal to $(p[i], d_p[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{4} \cdot 1\right) S_i = \frac{S_i + 3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 23 (only $q[i], d[i], d_p[i]$ are known) In this case, the similar argument as in Case 21 can be applied. Thus, $S_{i+1} = \frac{S_i + 3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 24 (only $q[i], d[i], d_q[i]$ are known) In this case, the similar argument as in Case 20 can be applied. Thus, $S_{i+1} = \frac{S_i+3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 25 (only $q[i], d_p[i], d_q[i]$ are known) In this case, the similar argument as in Case 22 can be applied. Thus, $S_{i+1} = \frac{S_i+3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 26 (only $d[i], d_p[i], d_q[i]$ are known) In this case, we first substitute the values of $d[i], d_p[i]$ and $d_q[i]$ for z_i, u_i and v_i in the second, the third and the fourth equations of (5) and check if all the equations of (5) are compatible. Clearly, if (x, y, z, u, v) is a *good* solution, these equations are compatible. Otherwise, from Conjecture above, these equations are compatible with the probability 1/4. Also, if these equations are compatible, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (z_i, u_i, v_i) should be equal to $(d[i], d_p[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i-1}{S_i} \cdot \frac{1}{4} \cdot 1\right) S_i = \frac{S_i+3}{4}$ with the probability $\delta^3(1-\delta)^2$.

Case 27 (only $d_q[i]$ is unknown) In this case, we first substitute the values of $p[i], d[i], q[i], d_p[i]$ for x_i, y_i, z_i, u_i in the first, the second and the third equations of (5) and check if these equations hold. Clearly, if (x, y, z, u, v) is a *good* solution, these equations hold with these values. Otherwise, from Conjecture above, these equations hold with the probability 1/8. Also, if these equations hold, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, y_i, z_i, u_i) should be equal to $(p[i], d[i], q[i], d_p[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i-1}{S_i} \cdot \frac{1}{8} \cdot 1\right) S_i = \frac{S_i+7}{8}$ with the probability $\delta^4(1-\delta)$.

Case 28 (only $d_p[i]$ is unknown) In this case, the similar argument as in Case 27 can be applied. Thus, $S_{i+1} = \frac{S_i+7}{8}$ with the probability $\delta^4(1-\delta)$.

Case 29 (only $d[i]$ is unknown) In this case, we first substitute the values of $p[i], q[i], d_p[i], d_q[i]$ for x_i, y_i, u_i, v_i in the first, the third and the fourth equations of (5) and check if these equations hold. Clearly, if (x, y, z, u, v) is a *good* solution, these equations hold with these values. Otherwise, from Conjecture above, these equations hold with the probability 1/8. Also, if these equations hold, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, y_i, u_i, v_i) should be equal to $(p[i], q[i], d_p[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i-1}{S_i} \cdot \frac{1}{8} \cdot 1\right) S_i = \frac{S_i+7}{8}$ with the probability $\delta^4(1-\delta)$.

Case 30 (only $q[i]$ is unknown) In this case, we first substitute the values of $p[i], d[i], d_p[i], d_q[i]$ for x_i, z_i, u_i, v_i in all the equations of (5) and check if the first, the second and the fourth equations are compatible and the third equation holds. Clearly, if (x, y, z, u, v) is a *good* solution, the first, the second and the fourth equations are compatible and the third equation holds with these values. Otherwise, from Conjecture above, the first, the second and the fourth equations are compatible and the third equation holds with the probability 1/8. Also, if the first, the second and the fourth equations are compatible and the third equation holds, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where (x_i, z_i, u_i, v_i) should be equal to $(p[i], d[i], d_p[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i-1}{S_i} \cdot \frac{1}{8} \cdot 1\right) S_i = \frac{S_i+7}{8}$ with the probability $\delta^4(1-\delta)$.

Case 31 (only $p[i]$ is unknown) In this case, the similar argument as in Case 30 can be applied. Thus, $S_{i+1} = \frac{S_i+7}{8}$ with the probability $\delta^4(1-\delta)$.

Case 32 (all $p[i], q[i], d[i], d_p[i], d_q[i]$ are known) In this case, we first substitute $p[i], q[i], d[i], d_p[i], d_q[i]$ for x_i, y_i, z_i, u_i, v_i in the equations of (5) and check if these equations hold. Clearly, if (x, y, z, u, v) is a *good* solution, these equations hold with these values. Otherwise, from Conjecture above, these equations hold with the probability 1/16. Also, if these equations hold, we can find exactly one solution $(x_i, y_i, z_i, u_i, v_i)$ from $(x, y, z, u, v) \in W_i$, where $(x_i, y_i, z_i, u_i, v_i)$ should be equal to $(p[i], q[i], d[i], d_p[i], d_q[i])$. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{16} \cdot 1\right) S_i = \frac{S_i + 15}{16}$ with the probability δ^5.

Summing up all the cases considered above, we have: for $i \geq 1$

$$S_{i+1} = \alpha S_i + \beta$$

for $\alpha = 2 - 5\delta + 5\delta^2 - \frac{5}{2}\delta^3 + \frac{5}{8}\delta^4 - \frac{1}{16}\delta^5, \beta = \delta + \delta^2 - \frac{3}{2}\delta^3 + \frac{3}{8}\delta^4 + \frac{1}{16}\delta^5$ and $S_1 = 1$. Consequently, we can get: for $i \geq 1$

$$S_i = \begin{cases} (\alpha + \beta - 1)\frac{1 - \alpha^{i-1}}{1 - \alpha} + 1 & \text{if } \alpha \neq 1 \\ 1 + (i-1)\beta & \text{if } \alpha = 1 \end{cases}.$$

At this point, it should be emphasized that the formula of S_i above is describing the average behavior of S_i since it involves the probabilistic value δ.

To analyze the meaning of the formula, we first note that, if $\alpha > 1$, then S_i will grow exponentially as i grows, thus, when implementing Algorithm 1, the variable W_i could hold exponentially many elements. Thus, in this case, the success of Algorithm 1 will depend on the computational environment. On the other hand, if $\alpha \leq 1$, the growth of S_i is at most linear in i, thus it is expected that Algorithm 1 will terminate in polynomial time in $n \approx \log_2 N$. More precisely, we have the following theorem.

Theorem 1 Under Conjecture above, the following holds. Let (N, e) be an RSA public key with $n = \lceil \log_2 N \rceil$. Further, assume that the δ-fraction of (p, q, d, d_p, d_q) is given. Then, if $\delta \geq 0.258$, Algorithm 1 can recover (p, q, d, d_p, d_q) in the expected polynomial time in n and e. Thus, if e is sufficiently small, Algorithm 1 will terminate in expected polynomial time in n.

Proof As noted in [5], finding k, k_p, k_q satisfying (2) requires $O(e)$ trials. And, if k, k_p, k_q are known, the running time of Algorithm 1 will be dominated by the loop size n and S_i at Step 2. In total, the expected running time of Algorithm 1 is $O(e \sum_{i=1}^{n} S_i)$ or $O(e\alpha^n)$. Thus, noting that, if $\delta \geq 0.258$, then $\alpha \leq 1$, Algorithm 1 can recover (p, q, d, d_p, d_q) in expected polynomial time in n and e. ■

We can obtain more implication from the formula of S_i. That is, if we are not insisting the key recovery algorithm to be polynomial-time but it is also acceptable that the computational environment can handle S_i, for example, up to 2^{30}, then the lower bound for δ in Theorem 1 can be reduced further. And, this may give an informal explanation about why Algorithm 1 still outputs the correct RSA private key in a reasonable time, even when δ is slightly less than 0.258. The more detail can be found in the next subsection.

A similar argument can be applied to the case (p, q, d) and we have the following algorithm.

Algorithm 2 (Erasure Correction for (p, q, d))

Input : (N, e) and δ-fraction of (p, q, d)

Output : (p, q, d)

1. $W_1 \leftarrow \{(1,1,1)\}$
2. For $i = 1$ to $n - 1$
   A. For each $(x, y, z) \in W_i$, find two solutions (x_i, y_i, z_i) of the corresponding equations in (5). And, if all x_i, y_i, z_i match with the corresponding known bits of p, q, d then add it to W_{i+1}. Otherwise, discard it.
3. For each $(x, y, z) \in W_n$, if $xy = N$, then output (x, y, z).

To estimate S_{i+1} from S_i, we should consider 8 cases, according to whether $p[i], q[i], d[i]$ are known or not.

Case 1 (all $p[i], q[i], d[i]$ are unknown) In this case, we can find exactly 2 extended solutions (x_i, y_i, z_i) from $(x, y, z) \in W_i$. Thus, $S_{i+1} = 2S_i$ with the probability $(1 - \delta)^3$.

Case 2 (only $p[i]$ is known) In this case, we can find one extended solution (x_i, y_i, z_i) from $(x, y, z) \in W_i$, where x_i should be equal to $p[i]$. Thus, $S_{i+1} = S_i$ with the probability $\delta(1 - \delta)^2$.

Case 3 (only $q[i]$ is known) In this case, the similar argument as in Case 2 can be applied. Thus, $S_{i+1} = S_i$ with the probability $\delta(1 - \delta)^2$.

Case 4 (only $d[i]$ is known) In this case, the similar argument as in the (p, q, d, d_p, d_q) case can be applied. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 2 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 2\right) S_i = S_i + 1$ with the probability $\delta(1 - \delta)^2$.

Case 5 (only $p[i], q[i]$ are known) In this case, the similar argument as in the (p, q, d, d_p, d_q) case can be applied. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1 - \delta)$.

Case 6 (only $p[i], d[i]$ are known) In this case, the similar argument as in the (p, q, d, d_p, d_q) case can be applied. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability $\delta^2(1 - \delta)$.

Case 7 (only $q[i], d[i]$ are known) In this case, the similar argument as in Case 6 can be applied. Thus, $S_{i+1} = \frac{S_i + 1}{2}$ with the probability $\delta^2(1 - \delta)$.

Case 8 (all $p[i], q[i], d[i]$ are known) In this case, the similar argument as in the (p, q, d, d_p, d_q) case can be applied. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{4} \cdot 1\right) S_i = \frac{S_i + 3}{4}$ with the probability δ^3.

And, we have the following theorem. \newline

Theorem 2 Under Conjecture above, the following holds. Let (N, e) be an RSA public key with

$n = \lceil \log_2 N \rceil$. Further, assume that the δ-fraction of (p, q, d) is given. Then, if $\delta \geq 0.412$, Algorithm 2 can recover (p, q, d) in the expected polynomial time in n and e. Thus, if e is sufficiently small, Algorithm 2 will terminate in expected polynomial time in n.

Proof Let W_i be the set of solutions $(x, y, z) \in \boldsymbol{Z}^3$ such that

$$N = xy \bmod 2^i$$

$$ez = 1 + k(x-1)(y-1) \bmod 2^i$$

and let S_i be the size of W_i. Then, we can get: $i \geq 1$

$$S_i = \begin{cases} (\alpha + \beta - 1)\dfrac{1 - \alpha^{i-1}}{1 - \alpha} + 1 & \text{if } \alpha \neq 1 \\ 1 + (i-1)\beta & \text{if } \alpha = 1 \end{cases}$$

for $\alpha = 2 - 3\delta + \frac{3}{2}\delta^2 - \frac{1}{4}\delta^3, \beta = \delta - \frac{1}{2}\delta^2 + \frac{1}{4}\delta^3$. Finally, note that if $\delta \geq 0.412$, then $\alpha \leq 1$. ■

For the (p, q) case, we have the following algorithm. \newline

Algorithm 3 (Erasure Correction for (p, q))

Input : (N, e) and δ-fraction of (p, q)

Output : (p, q)

1. $W_1 \leftarrow \{(1,1)\}$
2. For $i = 1$ to $n - 1$
   A. For each $(x, y) \in W_i$, find two solutions (x_i, y_i) of the corresponding equations in (5). And, if all x_i, y_i match with the corresponding known bits of p, q then add it to W_{i+1}. Otherwise, discard it.
3. For each $(x, y) \in W_n$, if $xy = N$, then output (x, y).

To estimate S_{i+1} from S_i, we should consider 4 cases, according to whether $p[i], q[i]$ are known or not.

Case 1 (all $p[i], q[i]$ are unknown) In this case, we can find exactly 2 extended solutions (x_i, y_i) from $(x, y) \in W_i$. Thus, $S_{i+1} = 2S_i$ with the probability $(1 - \delta)^2$.

Case 2 (only $p[i]$ is known) In this case, we can find one extended solution (x_i, y_i) from $(x, y) \in W_i$, where x_i should be equal to $p[i]$. Thus, $S_{i+1} = S_i$ with the probability $\delta(1 - \delta)$.

Case 3 (only $q[i]$ is known) In this case, the similar argument as in Case 2 can be applied. Thus, $S_{i+1} = S_i$ with the probability $\delta(1 - \delta)$.

Case 4 (all $p[i], q[i]$ are known) In this case, the similar argument as in the (p, q, d, d_p, d_q) case can be applied. Thus, $S_{i+1} = \left(\frac{1}{S_i} \cdot 1 + \frac{S_i - 1}{S_i} \cdot \frac{1}{2} \cdot 1\right) S_i = \frac{S_i + 1}{2}$ with the probability δ^2.

And, we have the following theorem. \newline

Theorem 3 Under Conjecture above, the following holds. Let (N, e) be an RSA public key with $n = \lceil \log_2 N \rceil$. Further, assume that the δ-fraction of (p, q) is given. Then, if $\delta \geq 0.585$, Algorithm 3 can recover (p, q) in the expected polynomial time in n and e. Thus, if e is sufficiently small, Algorithm 3 will terminate in expected polynomial time in n.

Proof Let W_i be the set of solutions $(x, y) \in \boldsymbol{Z}^2$ of $N = xy \bmod 2^i$ and let S_i be the size of W_i. Then, we can get: $i \geq 1$

$$S_i = \begin{cases} (\alpha + \beta - 1)\dfrac{1 - \alpha^{i-1}}{1 - \alpha} + 1 & \text{if } \alpha \neq 1 \\ 1 + (i-1)\beta & \text{if } \alpha = 1 \end{cases}$$

for $\alpha = 2 - 2\delta + \frac{1}{2}\delta^2, \beta = \frac{1}{2}\delta^2 + \frac{1}{4}\delta^3$. Finally, note that if $\delta \geq 0.585$, then $\alpha \leq 1$. ■

3.1 Experimental Result

To justify the presented analysis, we performed extensive experiments for reconstructing the erased RSA private key for the form $(p, q, d, d_p, d_q), (p, q, d)$ and (p, q). However, since there were already published some experimental results for the case (p, q, d, d_p, d_q) in [5], we just present the result for (p, q, d) and (p, q).

The experiments were proceeded as follows: for a given value δ, we first randomly generated 512-bit distinct primes p, q and computed d and k with fixed $e = 65537$. Then, we randomly selected δ-fraction of the bit positions so that the bit values of p, q, d are assumed to be given in those positions. After trying to recover the key bits, this process was then repeated 100 times and enumerated how many candidates are outputted from Algorithm 2 and 3 (with the final loop index $n/4$. The experiment was then repeated for various δ-values around the theoretical bounds in Theorem 2 and 3. Table 1 and 2 show these results.

Table 1. Experimental Result for (p, q)

δ	mininum	median	maximum	failure
0.55	1	15	1,081,104	1
0.56	1	11.5	5,110	0
0.57	1	9	32,325	0
0.58	1	10	64,500	0
0.59	1	5.5	6,268	0
0.60	1	4	867,087	0

Table 2. Experimental Result for (p, q, d)

δ	mininum	median	maximum	failure

0.39	1	17	312,812	2
0.40	1	21	285,160	3
0.41	1	15	587,776	0
0.42	1	7	70,833	0
0.43	1	7.5	29,696	0
0.44	1	7	1,244	0

The tables show the minimum, the median and the maximum of $S_{n/4}$. Also, in the tables, the 'failure' means the number of algorithms' failure where the algorithms abruptly terminated without outputting the return value, maybe due to the memory shortage of our computing system.

From the tables, we can conclude that there is an obvious tendency that $S_{n/4}$ grows as the δ-value decreases, which was predicted by Theorem 2 and 3.

4. Conclusion

Contrary to the common belief, DRAM which is used for the main memory of various computing devices retains its data even when it is powered-off. Especially, the data-retaining time can increase if DRAM is cooled down. Hence, if DRAM is suddenly powered-off, it could hold the cryptographic keys and the cold boot attack tries to recover these keys. And, using this cold-boot attack scenario, Heninger and Shacham presented a polynomial-time attack which, given a 0.27-fraction of the RSA private key bits of the form (p, q, d, d_p, d_q), can recover the whole RSA private key, provided that the given bits are uniformly distributed over the private key elements. And, based on their work, this paper presents a different approach for recovering the RSA private key bits from the decayed key image, under the assumption that some random portion of the private key bits is known. More precisely, we present an algorithm of recovering the RSA private key bits from the erased key material and elaborate the general formula of describing the number of partially-recovered RSA private key candidates in terms of the given erasure rate. Then, the result is justified with some extensive experiments.

References

[1] M. Albrecht and C. Cid, "Cold Boot Key Recovery by Solving Polynomial Systems with Noise," in *Proc. ACNS 2011*, pp. 57-72, June 7-10, 2011.
DOI: http://dx.doi.org/10.1007/978-3-642-21554-4_4

[2] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," *Journal of Cryptology*, Vol. 10, No. 2, pp. 233-260, 1997.
DOI: http://dx.doi.org/10.1007/s001459900030

[3] J.A. Halderman, S. Schoen, N. Heninger, W. Clarkson, W. Paul, J. Calandrino, A. Feldman, J. Appelbaum, and E. Felten, "Lest We Remember: Cold Boot Attacks on Encryption Keys," in *Proc. of USENIX Security 2008*, pp. 45-60, June 22-27, 2008.
DOI: http://dx.doi.org/10.1145/1506409.1506429

[4] W. Henecka, A. May and A. Meurer, "Correcting Errors in RSA Private Keys," in *Proc. CRYPTO '10*, pp. 351-369, Aug. 15-19, 2010.
DOI: http://dx.doi.org/10.1007/978-3-642-14623-7_19

[5] N. Heninger and H. Shacham, "Reconstructing rsa private keys from random key bits," in *Proc. CRYPTO 2009*, pp. 1-17, Aug. 16-20, 2009.
DOI: http://dx.doi.org/10.1007/978-3-642-03356-8_1

[6] A.A. Kamal and A.M. Youssef, "Applications of SAT Solvers to AES key Recovery from Decayed Key Schedule Images," in *Proc. SECURWARE 2010*, Jul. 18-25, 2010.
DOI: http://dx.doi.org/10.1109/SECURWARE.2010.42

[7] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," in *Proc. CRYPTO '99*, pp. 388-397, Aug. 15-19, 1999.
DOI: http://dx.doi.org/10.1007/3-540-48405-1_25

[8] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. CRYPTO '96*, pp. 104-113, Aug. 18-22, 1996.
DOI: http://dx.doi.org/10.1007/3-540-68697-5_9

[9] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[10] K.G. Paterson, A. Polychroniadou, and D.L. Sibborn, "A Coding-Theoretic Approach to Recovering Noisy RSA Keys," in *Proc. ASIACRYPT 2012*, pp. 386-403, Dec. 2-6, 2012.
DOI: http://dx.doi.org/10.1007/978-3-642-34961-4_24

[11] RSA Security INc., *Public-Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Standard,* 2002.

[12] A. Tsow, "An Improved Recovery Algorithm for Decayed AES Key Schedule Images," in *Proc. SAC 2009*, pp. 215-230, Aug. 13-14, 2009.
DOI: http://dx.doi.org/10.1007/978-3-642-05445-7_14

[13] J. Park and W. Choi, "Study on Structural and Systematic Security Threats of Vehicle Black Box as Embedded System," *International Journal of Advanced Culture Technology (IJACT)*, Vol. 9, No. 3, pp. 9-16, 2017.