2006

Server-aided public key generation protocols on low-power devices for ad-hoc networks

Yun Chen

Chengdi University of Information Technology

Weijian Chen

UEST of China

Yi Mu

University of Wollongong, ymu@uow.edu.au

Xin Chen

Zhengzhou University of Light Industry

Min Liu

UEST of China

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>

Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Chen, Yun; Chen, Weijian; Mu, Yi; Chen, Xin; and Liu, Min: Server-aided public key generation protocols on low-power devices for ad-hoc networks 2006.
<https://ro.uow.edu.au/infopapers/2932>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Server-aided public key generation protocols on low-power devices for ad-hoc networks

Abstract

Against the problems emerged in protocols presented by N. Modadugu, D.Boneh and M. Kim (2000), new server-aided public key generation protocols on low-power devices are proposed in this paper. The protocols are secure against some passive attacks, such as "collusion attack" and "the third party attack". In addition, efficiency of primality test on low-power devices is improved in this paper. Prospective applications of the protocols would be found in mobile e-commerce, mobile ad-hoc, and military communications.

Disciplines

Physical Sciences and Mathematics

Publication Details

Chen, Y., Chen, W., Mu, Y., Chen, X. & Liu, M. (2006). Server-aided public key generation protocols on low-power devices for ad-hoc networks. In G. Wen, S. Komaki, P. Fan & G. Landrac (Eds.), 2006 6th International Conference on ITS Telecommunications Proceedings (pp. 710-714). USA: IEEE.

Server-Aided Public Key Generation Protocols on Low-power Devices for Ad-hoc Networks

Yun Chen¹

Weijian Chen²

Yi Mu³

Xin Chen⁴

Min Liu²

¹School of Network Engineering, Chengdu University of Information Technology,
No.24, Block 1, Xue Fu Road, Chengdu, 610225, PR China.

²School of Communications and Information Engineering, UEST of China, Chengdu 610054, PR China

³School of Information Technology and Computer Science, University of Wollongong,
Northfields Avenue Wollongong, NSW 2522, Australia

⁴Institute of International Education, Zhengzhou University of Light Industry, Zhengzhou, 450002 P R China

Abstract—Against the problems emerged in protocols presented by N. Modadugu, D.Boneh and M. Kim [1], new server-aided public key generation protocols on low-power devices are proposed in this paper. The protocols are secure against some passive attacks, such as “collusion attack” and “the third party attack”. In addition, efficiency of primality test on low-power devices in reference [1] is improved in this paper. Prospective applications of the protocols would be found in mobile e-commerce, mobile Ad-hoc, and military communications.

I. INTRODUCTION

As a new application for in-vehicle entertainment, MP3 players have begun to make an entrance into the vehicle market. Connections compatible with suchlike devices (MP3) have been available on BMW, Nissan, Volvo, and so on. The logical extension of this trend is in-vehicle hard drivers and wireless local area networks (WLANs), which would permit the information exchange among home, office and vehicle via wireless connections—the storage and transfer of every sort of computer files such as databases, documents, music files, video files, and entertainments. Consumers would not drive to a video store to rent a movie, but would merely download the movie file via wireless connection from a convenient location—perhaps a metro station—and also pay in mobile devices such as mobile set, PDA, and other handheld devices, via the wireless connection. Most of these applications require secure means of communication such as encryption, authentication or digital signature schemes to perform their tasks without compromising confidentiality or authenticity. Generally, public key cryptosystems, such as RSA, or distribution system, such as Diffie-Hellman(D-H) algorithm, is absolutely necessary for realizing such schemes. Most of public key cryptosystems or distribution system deals

with modular exponentiation over finite field. To construct such systems, one usually needs to generate large primes. However, this causes a large amount of computations due to a computationally heavy primality test algorithm.

Initially, the public key was generated by a trusted server or a key management centre, and then issued to a low-power device offline. In Ad-hoc networks, however, there is no key management centre. Users (include low-power device holders) freely access in and out of the networks, they have to generate their own key on their own device. The most expensive time consuming computation in public key generation, for not only RSA but public key systems else such as ElGamal, DSA and D-H, is prime search. For low-power devices with only limited computational resources, the workload may be very heavy. As an example, it was reported that the generation of a 1024-bit RSA key took about 15 minutes on the PalmPilot [2].

N. Modadugu, D.Boneh and M. Kim [1] resolved the problem by presenting two-server-aided RSA key generation protocols, which we call “MBK”. Using the MBK protocols, users can quickly generate an RSA key on a low-power device with the help of *un-trusted* servers in such a way that once the key is generated the server should not get any information about the key it helped generation. However, one drawback of the MBK protocols is that two servers involved in generation of RSA key *must not* share information with each other. Otherwise, they can easily acquire the user’s secret key from share information as noted in [1].

In this paper, we present new version of the MBK protocols that resists a “collusion attack” in which the two servers in the server-aided public key generation collude to share information. Our protocols can also protect the low-power device users from “the third party attack” in which an eavesdropper hijacks all of the information of the user. We take RSA for example but not limit to RSA. The New

protocols are suitable for those systems based on modular exponentiation over finite field. Besides, our solution is almost as efficient as the original MBK protocols on low-power device and hence can be widely used in practice.

II. OVERVIEW OF RSA KEY GENERATION

In this section we give a brief overview of the RSA key generation. As a result of the RSA key generation, a key $\mathcal{RN}, \psi$ $e, \psi d \top \leftarrow$ can be generated. Here $N \psi = pq \psi$ denotes an n -bit modulus, where $p \psi$ and $q \psi$ are two large primes, each of which is $\frac{n}{2}$ bits long. $e \psi$ denotes the public exponent and $d \psi$ denotes the private exponent. A high-level description of the RSA key generation algorithm is as follows.

1) Repeat the following steps until two primes $p \psi$ and $q \psi$ are found:

- (a) Pick a random 512-bit candidate value $p \psi$ or q .
- (b) Check that $p \psi$ is not divisible by any small primes (i.e. 2, 3, 5, 7, etc.) using trial division.
- (c) Pick $g \in [2, p-1] \psi$ at random and check whether $g^{\frac{p-1}{2}} \equiv \pm 1 \pmod{p}$ or $g^{\frac{q-1}{2}} \equiv \pm 1 \pmod{q}$ (1) holds or not. (Note that all primes will pass this test while composites will fail with overwhelming probability).

2) Compute $N \psi = pq$.

3) Pick encryption and decryption exponents $e \psi$ and $d \psi$ such that $ed \equiv 1 \pmod{\varphi(N)}$, where $\varphi(N) = (p-1)(q-1)$. (Note that $\varphi \psi$ denotes the Euler function).

In the above algorithm, the most resource-consuming operations occur in Step 1(c), which is the primality testing part. Compared with 1(c), computational overhead for conducting Step 2 and Step 3 are small. We remark that when RSA cryptosystem is used to encrypt a short session key (for a symmetry encryption scheme), which is shorter than the bit-length of p , one can use the “unbalanced RSA”, firstly considered by Shamir [3]. In his unbalanced RSA, the encryption and decryption exponent $e \psi$ and $d \psi$ satisfy $ed \equiv 1 \pmod{(p-1)}$ and the size of a prime factor $q \psi$ is approximately 10 times larger than the other prime factor p . Shamir [3] noted that the key generation of the standard RSA can be sped up in the unbalanced version without extra time complexity. In [1], N. Modadugu et al. present a variant of the unbalanced RSA where a modulus is a product of a prime and a certain size of random number. It is evident that in this variant, the computations in Step 1(c) can be greatly reduced due to the fact that only one prime needs to be generated. (As a result, the RSA key generation in [2] can be sped up further).

III. THE MBK PROTOCOLS

In this section we review the MBK protocols for standard and unbalanced RSA key generation.

A. Standard RSA key generation

The main idea of MBK protocol to generate a standard

RSA modulus is to divide quarter of candidate $\frac{\varphi(N)}{4} = \frac{N-p-q+1}{4}$ into two parts: t_1 and t_2 such that $t_1 + t_2 = \frac{\varphi(N)}{4}$ and then ask two un-trusted servers for assistance to generate key. We describe the protocol as follows:

1) Low-power device generates two candidates $p \psi$ and $q \psi$ such that $p \equiv q \equiv 3 \pmod{4}$, each of them is $\frac{n}{2}$ -bit in length; and none of them is divisible by small primes.

2) Low-power device computes $N = pq$ and $\varphi(N) = N - p - q + 1$. At the moment, $\varphi(N)$ is just a candidate, not an Euler number. Then pick a random $g \in Z_N^*$ such that g is not a multiple of p or q .

3) Low-power device picks random integers t_1 and t_2 in $(-N, N)$ such that $t_1 + t_2 = \frac{\varphi(N)}{4}$. Then the low-power device sends $\langle N, g, t_1 \rangle$ to server 1 and $\langle N, g, t_2 \rangle$ to server 2 respectively.

4) Server 1 calculates $X_1 \equiv g^{t_1} \pmod{N}$ and server 2, $X_2 \equiv g^{t_2} \pmod{N}$ separately. Both results are sent back to low-power device.

5) Low-power device judges if N is a potential RSA modulus or not by checking $X_1 \equiv \pm X_2 \pmod{N}$.

6) Low-power device locally runs a probabilistic primality test once to verify that servers returned correct values. [1]

B. Unbalanced RSA key generation

So-called unbalanced RSA means that RSA modulus is a product of a prime p and a random number or a longer prime which is 8-10 times long as the prime p in bits.

The main idea of the MBK protocol for unbalanced RSA is similar to that for standard RSA. The differences are: (a) Low-power device divides $(p-1)$ (not $\frac{\varphi(N)}{4}$) into two parts such that $t_1 + t_2 = p-1$. (b) The low-power device checks if $X_1 \equiv \pm X_2 \pmod{p}$ (The modulus is not N).

In the above protocol, the total running time is reduced by a factor of 5 [1]. Here, it is worth emphasizing that an unbalanced RSA is only suitable for the case when the size of a plain-text is smaller than that of prime p . Besides, the unbalanced RSA key can only be used for encryption. It cannot be used for signature.

IV. COUNTERMEASURES: TWO NEW SERVER-AIDED RSA KEY GENERATION PROTOCOLS SECURE AGAINST SOME PASSIVE ATTACKS

Recall that in MBK protocols, a low-power device generates its own key with the help of two un-trusted servers without compromising the security of the key if and only if the servers cannot exchange information with each other. Actually, it is very difficult to isolate two servers over public networks. Once two key generation servers collude, it is easy for them to obtain the hand-held device's private key. More precisely, if the two servers

exchange t_1 and t_2 in the standard RSA version of the MBK protocol, they can easily recover $\varphi(N)$ by simply computing $\varphi(N) = 4(t_1 + t_2)$. Furthermore, they figure out the user's private key $d\psi$ from $\varphi(N)$ and public key $e\psi$ by running the Euclidean algorithm [5] to get $d \equiv e^{-1} \pmod{\varphi(N)}$. Moreover, in mobile communication environment, it is possible for an eavesdropper to hijack all of the information between the low-power device and the servers involved in computation. Hence the eavesdropper can figure out the user's private key from the intercepted information. In this section, we present two new server-aided RSA key generation protocols secure against the above passive attacks.

A. Standard RSA key generation protocol secure against passive attack

A standard RSA key generation version of the new protocol can be described as follows:

1) Low-power device generates two candidates $p\psi$ and $q\psi$ such that $p \equiv q \equiv 3 \pmod{4}$, each of them is $\frac{n}{2}$ -bit in length, and none of them is divisible by small primes.

2) Low-power device computes $N=pq$ and $\varphi(N)=N-p-q+1$, where $\varphi(N)$ is just a candidate, not an Euler number. Then pick a random $g \in Z_N^*$ such that g is not a multiple of p or q .

3) Low-power device picks random integers t_1, t_1' and t_2, t_2' in $(-N, N)$ such that

$$t_1 + t_1' = t_2 + t_2' = \frac{\varphi(N)}{4} \quad (2)$$

4) Low-power device picks random integers k, l and large numbers a and b in $(0, \varphi(N))$ such that

$$\gcd(a, \varphi(N)) = 1, \quad \gcd(b, \varphi(N)) = 1 \quad (3)$$

Then compute

$$s_1 = [a + k(p-1)]t_1, \quad s_2 = -at_2 \quad (4)$$

$$s_1' = [b + l(q-1)]t_1', \quad s_2' = -bt_2' \quad (5)$$

5) Low-power device sends $\langle N, g, s_1, s_1' \rangle$ to server 1 and $\langle N, g, s_2, s_2' \rangle$ to server 2

6) Server 1 computes $X_1 \equiv g^{s_1} \pmod{N}$ and $X_1' \equiv g^{s_1'} \pmod{N}$. Server 2 computes $X_2 \equiv g^{s_2} \pmod{N}$ and $X_2' \equiv g^{s_2'} \pmod{N}$. All of the results are sent back to the low-power device.

7) Low-power device checks if $X_1 \equiv \pm X_2 \pmod{p}$, and $X_1' \equiv \pm X_2' \pmod{q}$. If so, p and q are considered to be potential primes. Then, $N=pq$ is declared as a potential RSA modulus. Otherwise, restart the operations from step 1).

8) Low-power device locally runs a probabilistic primality test with a small Mersenne prime m to verify that servers returned correct values.

Now, we verify the soundness of the above protocol.

For all $g \in (-N, N)$ and $\gcd(g, N) = 1$, according to Euler theory

$$g^{\varphi(N)} \equiv 1 \pmod{N} \quad (6)$$

When $p \equiv q \equiv 3 \pmod{4}$ holds, it follows that

$$g^{\frac{\varphi(N)}{4}} \equiv \pm 1 \pmod{N} \quad (7)$$

Namely,

$$g^{t_1} \equiv \pm g^{-t_2} \pmod{N} \quad (8)$$

Equation (3) reads that a is an odd, hence

$$g^{at_1} \equiv \pm g^{-at_2} \pmod{N} \quad (9)$$

According to number theory, we get following formulae from Equations (8) and (9) respectively

$$g^{t_1} \equiv \pm g^{-t_2} \pmod{p} \quad (10)$$

$$g^{at_1} \equiv \pm g^{-at_2} \pmod{p} \quad (11)$$

Combine step 6), part A, section IV with Equation (11), we have

$$\begin{aligned} X_1 &\equiv g^{s_1} \equiv g^{[a+k(p-1)]t_1} \equiv g^{at_1} g^{k(p-1)t_1} \equiv g^{at_1} \\ &\equiv \pm g^{-at_2} \equiv \pm g^{s_2} \equiv \pm X_2 \pmod{p} \end{aligned} \quad (12)$$

Where

$$(g^{p-1})^{kt_1} \equiv 1^{kt_1} \equiv 1 \pmod{p} \quad (13)$$

The current problem is whether or not Equation (10) holds when Equation (12) is true?

Let $j \equiv a^{-1} \pmod{p-1}$, consider precondition $\gcd(a, \varphi(N)) = 1$, and raise both sides of Equation (11) to j , then

$$(g^{at_1})^j \equiv (g^{aa^{-1}t_1})^j \equiv \pm (g^{-at_2})^j \equiv \pm (g^{-aa^{-1}t_2}) \pmod{p} \quad (14)$$

Where $g^{aa^{-1}} \equiv g \pmod{p}$. It follows that

$$g^{t_1} \equiv \pm g^{-t_2} \pmod{p}$$

Similarly, we get

$$X_1' \equiv \pm X_2' \pmod{q} \quad (15)$$

On the other hand, if $p\psi$ or $q\psi$ is not a prime, Equation (7) will fail with overwhelming probability. Consequently, Equations (12) and (15) will fail with overwhelming probability.

B. Unbalanced RSA key generation protocol secure against passive attack

We describe the unbalanced RSA key generation protocol against passive attack as follows:

1) Low-power device generates a 512-bit candidate p such that $p \equiv 3 \pmod{4}$ and p is not divisible by small primes and a random number R that is 8-10 times long as p in bits, usually 4096-bit. Then compute $N = pR$

2) Low-power device picks a random integers t_1 and t_2 in $(-p, p)$ such that $t_1 + t_2 = \frac{p-1}{2}$. Then pick random integers a, k in $(0, N)$ such that

$$\gcd(a, p-1) = 1 \quad (16)$$

Then compute $s_1 = [a + k(p-1)]t_1$, $s_2 = -at_2$

3) Low-power device picks a random number $g \in Z_N^*$ such that g is not multiple either p or q . Then the device sends $\langle N, g, s_1 \rangle$ to server 1 and $\langle N, g, s_2 \rangle$ to server 2

4) Server 1 computes $X_1 \equiv g^{s_1} \pmod{N}$, server 2 computes $X_2 \equiv g^{s_2} \pmod{N}$. Both results are sent back to low-power device.

5) Low-power device checks if $X_1 \equiv \pm X_2 \pmod{p}$. If so, $N = pR$ is declared as a potential unbalanced RSA modulus. Otherwise, restart from step 1).

6) Low-power device locally runs a probabilistic primality test to verify that p is a prime.

In this version, the total running time is reduced by a factor of 5 [4]. It is worth noticing that unbalanced RSA is only suitable for the case that message is shorter than p . Besides, the version cannot be used for signatures.

C. Security analysis

Analysis in [1] showed that the un-trusted servers learnt no useful information about RSA private key whether from the rejected candidates or from the final public key if only server 1 did not collude with server 2. So is the case in our protocols. Also in our protocols, the un-trusted servers shall not get any useful information even if they share information with each other.

In the standard version of new protocol, server 1 learns only s_1 and s'_1 , server 2 knows only s_2 and s'_2 . They are not able to figure out t_1 and t_2 from s_1 and s_2 because they do not know a, k , and prime p , or b, l , prime q . Suppose the two servers collude and calculate

$$4(s_1 - s_2) = (p-1)[a(q-1) + 4kt_1], \quad (17)$$

the un-trusted servers have to factorize great composite if they want to derive $(p-1)$ from equation (17). As we know, it is computationally infeasible as long as the integer is greater than 576 bits [4]. Similarly, there is no way for them to figure out $(q-1)$ from s'_1 and s'_2 . Hence the servers are not able to get $\varphi(N)$. In result, the servers shall know nothing about user's private key even if they could share information with one another.

It is also impossible for any attacker to get p or q or $\varphi(N)$ by random guessing when N , s_1 and s_2 are large enough.

Therefore, users are able to generate RSA key with the assistance of two un-trusted servers on low-power device against some passive attacks, such as "collusion attack" and "the third party attack", in our new protocols.

With a similar argument, the unbalanced RSA version of our protocol can be proven to be secure against some passive attacks

It is the very reason that we can do single server-aided key generation without extra work added on low-power device.

V. CONCLUDING REMARKS

We substitute random number g with small Mersenne prime m to do the primality test in our protocols. The reasons are: (a) Small Mersenne prime must satisfy $\gcd(m, N) = 1$. (b) Multiplication by m can be substituted with simply shift and addition. It is obvious that the primality test in our approaches is faster than that in MBK protocols.

The MBK protocols efficiently solved the problem of RSA key generation on a low-power device with the help of two un-trusted servers. However, the protocols are insecure against collusion attack and eavesdrop.

In this paper, we present new protocols in which the problem of some passive attacks are successfully solved by hiding useful information t_1, t'_1 and t_2, t'_2 , respectively behind s_1, s'_1 and s_2, s'_2 .

There are small works added to a low-power device in our protocols. For legal user, it is easy to check the validity of the candidates. For malicious servers or the third party attackers, they must factorize very large composite if they want to filch user's private information.

Another merit is that the new protocols are also suitable for the applications of single server-aided key generation without causing extra workload on low-power device. Note that as in discussed in [6], computing power of low-power device is much smaller than that of servers. From this point of view, our protocols are preferable to the MBK protocols for single server-aided key generation case, in which low-power device must compute modular exponentiation $g^r \pmod{N}$ or $g^r \pmod{p}$, where r is 160 bits long integer.

In addition, the holder of a low-power device need not worry about eavesdropping on network from the third party. While using the MBK protocols, we must make special protection measure between low-power device and server in order to protect an eavesdropper from overhearing.

However, large amount of work are added to server 1 and 2, the length of exponent dominates the speed of key generation. In NBK protocols, exponents t_1 and t_2 are similar length to $\frac{\varphi(N)}{4}$, while s_1 and s_2 are much longer than that of $\varphi(N)$ in the new protocol, the maximum length can reach nearly two times of $\varphi(N)$'s. that may slow down the working speed one multiple [2]. In other words, we acquire security key generation in the cost of working speed of the servers. Nevertheless it is still much faster than local generation on low-power device according to reference[4].

Of course, we should take notice when we chose parameters of unbalanced RSA key [3].

We take RSA for example of server-aided key generation on low-power device, but not limit to RSA. The approach is also suitable for those public key systems based on modular exponentiation over finite field such as DSA, D-H, ElGamal, and so on.

VI. OPEN PROBLEM

It remains an open question: whether there is any way else to break our protocols?

REFERENCES

- [1] N. Modadugu, D. Boneh, and M. Kim, "Generating RSA Keys on a Handheld Using an Untrusted Server," In CT-RSA 2000, Available at <http://crypto.stanford.edu/dabo/pubs.html>.
- [2] D. Boneh and M. Franklin, "Efficient Generation of Shared RSA Keys", Journal of the ACM, 48(4):702–5 722, 2001.
- [3] A. Shamir, "RSA for Paranoids", CryptoBytes, 1(3):1–4, 1995.
- [4] RSA Security, "RSA-576 is Factored!", Available at <http://www.rsasecurity.com/rsalabs/node.asp?id=2096>. January, 2005
- [5] H.Cohen, "A Course in Computational Algebraic Number Theory", Springer-Verlag: Vol.138 of Graduate Texts in Mathematics, 1996
- [6] G.Horng, "A Secure Server-aided RSA Signature Computation Protocol for Smart Cards", Journal of Information Science and Engineering, 16(6):847-855, 2003.6