Backdoors in Cryptographic Key Generators

Yohan Chenebault, Thomas Hugueville
ENSEEIHT
France

ABSTRACT

Cryptographic protocols are used everywhere and their implementations are often trusted blindly by users. However the presence of backdoors is a possibility that should not be forgotten. In this paper we discuss asymetric backdoors and analyse our implementation of a backdoored Diffie-Hellman key generator in the OpenSSL cryptographic library.

Keywords

cryptography, backdoor, OpenSSL, kleptography

1. INTRODUCTION

Cryptography is the science of secret. It allows people to communicate over a unsecured channel while guaranteeing the secrecy, integrity and authenticity of the messages. Most of the research is aimed towards finding flaws and vulnerabilities in either the design of cryptosystem or their implementations. However, the study of backdoors – or kleptography – has been raising more and more interest over the past few years.

Kleptography is the art of stealing information subliminally and securely. It is for instance possible to implement a backdoor in such a way that the attacker can recover any private key through the public key without the user knowing.

2. STATE OF THE ART

One way to implement a backdoor is to compromise the underlying pseudorandom number generator (PRNG). Backdoors in PRNGs are very interesting because every cryptographic protocols rely on them. As such, they have a very large influence. However it requires the attacker to be able to obtain at least one output of the PRNG.

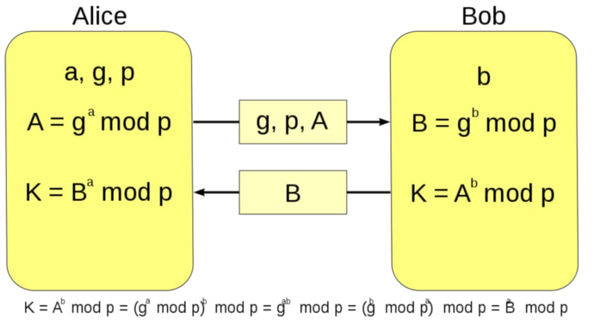
One example of a backdoor in a PRNG is the case of Dual_EC_DRBG. This algorithm was designed by the NSA in 2004 and was incorporated into the NIST, ANSI and ISO standards. The PRNG was believed to contain a backdoor. Indeed, it was proven that it was possible to implement a backdoor with it but it was not possible to prove that the NSA effectively backdoored it. However, Edward Snowden’s leaks in 2013 actually confirmed that Dual_EC_DRBG contained a backdoor. After that incident, further research was made on backdoors in PRNGs and it was proven that one could design a backdoored PRNG in such a way that the knowledge of one output allowed the attacker to recover every past and future outputs. Furthermore, some designs allows even PRNG with inputs, i.e. that are reseeded on regular basis, to be vulnerable to backdoors. In that case it is possible to recover every past outputs thanks to a single outputs

Another way is to implement backdoors is in the key generation algorithm. In that case, the idea is to encode the private information into the public information. For instance it is possible to create a backdoored RSA so that the attacker can recover the private key with only the knowledge of the public key and their own private key.

A primitive for discrete-log cryptosystems (Diffie-Hellman, DSA, ElGamal, …) exists as well. Those designs guarantee indistinguishibility, confidentialy, forward secrecy and indetectability.

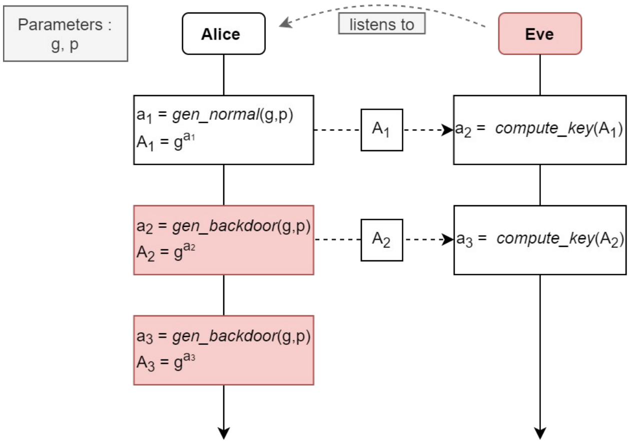
3. DIFFIE-HELLMAN KEY EXCHANGE

Diffie-Hellman (DH) is a key exchange algorithm based on the discrete log problem. To obtain a shared secret, Alice and Bob agree on DH parameters: a prime *p* and a generator *g* of a subgroup of $\mathbb{Z}_p$. Each of them generate a random private key (*a* and *b*) then compute their public keys $A=g^a \, mod \, p$ and $B=g^b \, mod \, p$. The shared secret is then $K=g^{ab} \, mod \, p$.


Alice
Bob
a, g, p
A = g^a mod p
K = B^a mod p
g, p, A
B
b
B = g^b mod p
K = A^b mod p
K = A^b mod p = (g^a mod p)^b mod p = g^ab mod p = (g^b mod p)^a mod p = B^a mod p

4. THE BACKDOOR IN DIFFIE-HELLMAN

The attacker possesses their own DH key pair (e,E) with $E \equiv g^e \, mod \, p$. The attack is first initialised with a normally randomly generated key *a₁*. This key is stored in memory and will be used to compute the following key. The compromised key is $a_2=PRF(E^{a_1})$ where *PRF* is a pseudorandom fuction. The attacker is then able to retrieve *a₂* thanks to the relation $A_1^e \equiv g^{a_1 e} \equiv E^{a_1} \, mod \, p$.



5. SECURITY OF THE ATTACK

This design guarantees both the indistiguishibility of the outputs and the confidentiality of the outputs against an adversary that would have reverse-engineered the system.

However, a timing analysis reveals a flaw in our implementation: the compromised key generation is 50% slower than the normal key generation. This behaviour may alert the user of the presence of the backdoor. To correct this issue, we have to introduce a time buffer such that both operations take the same amount of time.

6. CONCLUSIONS AND FUTURE APPLICATIONS

We saw that backdoors could be implemented in various ways. More importantly, because they can be designed in such a way that they cannot be detected we believe that they are a field worth investigating and studying so as to be more aware of them and how to counteract them (which is mostly vigilance).

We were able to implement our own version of a backdoored DH into OpenSSL that can only be exploited by the attacker. Such a backdoor could deployed through a virus, by usurping a depot or simply by having access to a target host once.

7. ACKNOWLEDGEMENTS

We would like to thank Vincent Migliore, our professor of cryptography and supervisor for this project.

8. REFERENCES

[1] Dan Shumow and Nils Ferguson, On the possibility of a back door in the NISTSP800-90 Dual EC PRNG.
[2] Jean Paul Degabriele et al., Backdoors in Pseudorandom NumberGenerators: Possibility and Impossibility Results
[3] Adam Young et Moti Yung, Malicious Cryptography: Exposing Cryptovirology
[4] Adam Young et Moti Yung, http://www.cryptovirology.com/