A Fault-Injection Attack on Fiat-Shamir Cryptosystems

Artemios G. Voyiatzis and Dimitrios N. Serpanos
Computer Systems Laboratory
Department of Electrical and Computer Engineering
University of Patras
GR-26504 Rion Patras
Greece
{bogart,serpanos}@ee.upatras.gr

Abstract

Fault-injection attacks and cryptanalysis is a realistic threat for systems implementing cryptographic algorithms. We revisit the fault-injection attacks on the Fiat-Shamir authentication scheme, a popular authentication scheme for service providers like pay per view television, video distribution and cellular phones.

We present a new and effective attack on cryptosystems that implement the Fiat-Shamir identification scheme. The attack is successful against all system configurations in contrast to the original Bellcore attack, which has been proven incomplete (easy to defend against).

1. Introduction

System parameters measured during execution of cryptographic algorithms can be exploited by an attacker in order to discover the secret keys used [11] [6]. The field of *implementation* or *side-channel* cryptanalysis has drawn significant attention by the security community. In contrast to classical (mathematical) cryptanalysis, implementation cryptanalysis targets implementations of cryptographic algorithms. Side-channels, not covered by the mathematical model of the algorithms, transmit to the environment information for the secret keys employed in a cryptographic operation. Appropriate analysis of this information can be utilized to extract the whole keys and thus render insecure the specific cryptographic system.

Implementation cryptanalysis can be categorized in two classes of attacks: passive and active. In passive attacks, the side-channel is a measurable parameter of the implementation. Examples include algorithm execution time [11], power consumption [12], and EM radiation [10], [1]. In all cases, the attacker does not physically alter the system under attack, but only collects information from it with external measurement equipment. In active (or fault-injection) attacks, the attacker injects hardware faults, such as flip bits in memory, which lead to undetectable erroneous output. Appropriate use of erroneous output by the attacker can lead to full disclosure of system secret keys. Fault injection in cryptographic devices can be realized, for example, by operation in extreme conditions, as described in [2],[9].

The applicability and practicality of implementation cryptanalysis and, especially, for the active attacks has been questioned [13]. Research results demonstrated that such attacks are indeed feasible [4]. Furthermore, the applicability of timing attacks (a case of passive attacks) was expanded to Internet systems and, more specifically, to secure web servers utilizing the OpenSSL cryptographic library [8]. Thus, implementation cryptanalysis is a real threat, and appropriate countermeasures must be employed to protect cryptosystems.

Most popular cryptographic algorithms have been shown vulnerable to fault-injection attacks: RSA using Chinese Remainder Theorem, RSA using Montogommery arithmetic, the Schnorr identification scheme and the Fiat-Shamir authentication scheme [6]; DES and other symmetric key cryptosystems [5]; and lately AES [9].

We revisit the active attack on the Fiat-Shamir authentication scheme. In [14], we proved that the *Bellcore attack* [6] is incomplete, since it is based on an assumption that does not always hold. Thus, we proved that there exist Fiat-Shamir systems configuration which defend the attack and resist such kind of cryptanalysis.

In this paper, we propose a new fault-injection attack, which is provably valid against all Fiat-Shamir configurations, under stronger assumptions. This new attack is not only successful but efficient and realistic for typical environments, such as smart cards.

The paper is organized as follows. Section 2 describes briefly the Fiat-Shamir identification scheme, the Bellcore attack and its fault insertion model, and the proposed de-

Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04)
0-7695-2087-1/04 $20.00 © 2004 IEEE

fense named *Precautious Fiat-Shamir scheme*. Section 3 presents a novel and effective fault-injection attack against Fiat-Shamir systems and proves its correctness.

2. Background

2.1. Fiat-Shamir identification scheme

The Fiat-Shamir identification scheme is a zero-knowledge scheme, where one party authenticates its identity to another using an asymmetric, public key method.

The scheme works as follows, assuming that Alice authenticates herself to Bob. Alice has as a public key the set $PK_\ell = \{u_i \mid u_i = s_i^2 \pmod N \text{ and } 1 \le i \le \ell\}$, where N (an n-bit modulus) is the product of two large prime numbers, and a set of invertible elements $s_1, s_2, \ldots, s_\ell \pmod N$. Alice proves her identity to Bob using the following protocol:

1. Alice and Bob agree on a security parameter, $t \le \ell$;
2. Alice chooses a random number $r \in \mathbb{Z}_N^*$, calculates $r^2 \bmod N$ and sends this number to Bob;
3. Bob chooses a random subset $S \subseteq \{1, \ldots, t\}$ and sends S to Alice;
4. Alice computes $y = r \cdot \prod_{i \in S} s_i \bmod N$ and sends y to Bob;
5. Bob verifies Alice's identity by verifying that:

$$y^2 = r^2 \cdot \prod_{i \in S} u_i \pmod N$$

The security of the scheme is based on the hypothesis that computation of square roots is a hard problem over $\mathbb{Z}_N$.

2.2. The Bellcore attack on Fiat-Shamir

The Bellcore attack [6] [7] is a theoretical active attack that exploits erroneous computations and derives secret keys for various cryptographic protocols. In the case of Fiat-Shamir, it derives the secret elements, $s_1, \ldots, s_\ell$.

Assuming that it is possible to introduce transient bit flips during Alice's computations, Bob implements an attack introducing bit flips in r, during Step 3 of the protocol described above, while Alice waits for Bob to send the subset S. This leads Alice to compute in Step 4 with an incorrect value of r. This, in turn, enables Bob to calculate Alice's secret elements, as we describe below.

Importantly, in this case, the attacker solves the time isolation problem [3], which constitutes a significant difficulty in the implementation of active attacks. Specifically, the attacker (Bob) does not need exact synchronization with the device that acts as Alice, because the attacker can delay transmission of the subset S arbitrarily. So, the attacker needs to solve only the space isolation problem [3], i.e., he needs to locate the correct memory location that stores r, in order to introduce the transient bit flip. For simplicity, we assume in the following that a single bit flip occurs.

The Bellcore attack on Fiat-Shamir identification scheme is summarized in the following theorem [6]:

Theorem 1 (Bellcore attack) *Let us consider an instance of the Fiat-Shamir scheme with N an n-bit modulus and ℓ the predetermined security parameter. Given ℓ erroneous executions of the protocol, an attacker can recover the secret $s_1, \ldots, s_\ell$ in the time required to perform $\mathrm{O}(n\ell + \ell^2)$ modular multiplications.*

Proof 1 (summarized) *A fault injection at position i of r, $i \in \{0, 1, \ldots, n-1\}$, implements a bit-flip, which changes the original value of r by adding the value E, where $E = \pm 2^i$ (the sign of the change depends on whether the bit-flip caused a 0-to-1 or a 1-to-0 bit-flip).*

When the bit-flip occurs, Alice calculates an incorrect value of y, denoted as $\hat{y}$, during Step 4 of the protocol and sends it to Bob:

$$\hat{y} = (r + E) \cdot \prod_{i \in S} s_i$$

So, Bob can compute

$$T(S) = \prod_{i \in S} s_i = \frac{2E \cdot \hat{y}}{\frac{\hat{y}^2}{\prod_{i \in S} u_i} - r^2 + E^2} \bmod N$$

Bob validates the correctness of his bit-flip guess by checking that

$$T^2(S) = \prod_{i \in S} u_i$$

Considering that Bob does not know the specific position i where the bit-flip occurred, this step requires $\mathrm{O}(n+\ell)$ modular multiplications, because Bob must try all possible bit error positions. Thus, for ℓ different sets S, the attack requires $\mathrm{O}(n\ell + \ell^2)$ modular multiplications.

Given a method to compute $T(S)$ for various sets S, an attacker needs an algorithm to derive each $s_1, s_2, \ldots, s_\ell$. If Alice accepts singleton sets, the algorithm is trivial: Bob chooses $S = \{k\}$ and then, $T(S) = s_k$. Thus, Bob needs only ℓ iterations to collect all ℓ possible s_i's.

If Alice does not accept singleton sets, Bob can use the following algorithm: Bob can map each set S to its characteristic binary vector $U \in \{0,1\}^\ell$, i.e. $U_i = 1$ if $i \in S$; if Bob can construct an $\ell \times \ell$ full rank matrix over $\mathbb{Z}_2$, then Bob can derive each s_i. For example, to determine s_1, Bob constructs elements $a_1, a_2, \ldots, a_\ell \in \{0,1\}$, so that

$$a_1 U_1 + \ldots + a_\ell U_\ell = (1, 0, 0, \ldots, 0) \pmod 2$$

This is efficient, because vectors $U_1, \ldots, U_\ell$ are linearly independent over $\mathbb{Z}_2$. When computations are made over

the integers, we have:

$$a_1U_1 + \ldots + a_\ell U_\ell = (2b_1 + 1, 2b_2, 2b_3, \ldots, 2b_\ell)$$

for some known $b_1, \ldots, b_\ell$*. Then, Bob calculates* s_1 *as:*

$$s_1 = \frac{T_1^{a_1} \cdots T_l^{a_\ell}}{u_1^{b_1} \cdots u_\ell^{b_\ell}} \pmod N$$

The calculation of s_1 *requires* $\mathrm{O}(\ell)$ *modular multiplications; so, the calculation of all* s_i*,* $i \in \{1, \ldots, \ell\}$ *requires* $\mathrm{O}(\ell^2)$ *modular multiplications.*

Overall, the cost of the algorithm is $\mathrm{O}(n\ell + \ell^2)$ *modular multiplications.*

2.3. The precautious Fiat-Shamir scheme

In [14], we formally proved that the Bellcore attack is not successful, in general, on systems that implement the Fiat-Shamir scheme, because it is based on an assumption which is not always true. Specifically, the construction of an $\ell \times \ell$ matrix with rank ℓ over $\mathbb{Z}_2$ is not always possible. Thus, if a device operating as Alice in the scheme can accept challenges from an attacker with specific requirements (so that the $\ell \times \ell$ matrix with rank ℓ is infeasible), the Bellcore attack is not successful, because it cannot derive any secret key from the erroneous output. Based on this fact, we introduced the Precautious Fiat-Shamir protocol, which proactively defends Bellcore attack.

The protocol changes just the third step (Step 3) of the protocol used in the Fiat-Shamir scheme, as follows:

Definition 1 *A Fiat-Shamir Identification Scheme augmented with a set* G *of even numbers is called* precautious*, if Alice accepts on the third step only* S*, such that* $|S| \in G$*.*

It should be noted that the proposed limitation of acceptable challenges does not decrease the security offered by the scheme.

3. A new attack on the Fiat-Shamir scheme

We present a novel fault-injection attack model which is successful against both the classical and Precautious Fiat-Shamir schemes. This model allows, in all cases, Bob to derive Alice's secret elements in polynomial time. Extracting each and every secret key of the device acting as Alice in the scheme, enables undetectable impersonation.

3.1. Fault injection model

We assume that the attacker is able to inject a transient fault in any of the registers holding the secret information $s_1, \ldots, s_\ell$ during the computations of Step 4 of the protocol. Our method can identify the s_i that was altered.

The error must be injected in the register holding an s_i before it is used to compute of the reply $r \prod_{i \in S} s_i$. For a successful attack, Bob (the attacker) needs to solve both time and space isolation problems in this case, because he cannot control this step of the protocol in time.

In this context, our assumption of the fault model is stronger than that of Bellcore's attack, because we need exact synchronization with the device that acts as Alice, while, in contrast, the Bellcore attack needs to solve only the space isolation problem.

For simplicity, we present the analysis for the case of a single bit flip. Similarly to the Bellcore attack, our model is effective for multiple bit flips, with increased complexity.

3.2. Fault-injection attack model

Using the predefined fault model, the new attack is described in Theorem 2.

Theorem 2 *Let us consider an instance of the Fiat-Shamir scheme with* N *an n-bit modulus and* ℓ *the predetermined security parameter. Given* ℓ *erroneous executions of the protocol, an attacker can recover the secret* $s_1, \ldots, s_\ell$ *in the time required to perform* $\mathrm{O}(n\ell^2)$ *modular multiplications.*

Proof 2 *Assume that a fault injection occurs in Step 4, during a protocol invocation, causing a bit-flip. During Step 5 of the protocol, Bob can detect that an error indeed occurred. Without loss of generality, we assume that the error occurred in* s_j*. Then, the attacker can derive* s_j *as follows.*

Since a single bit flip occurred, s_j *was changed in Step 4 to* $s_j \pm 2^i$*, for some* $0 \leq i \leq n-1$*. Thus, after such a protocol invocation, Bob has collected the following numbers (during the relative protocol steps):*

Step 1:

$$r_1^2 \pmod N$$

Step 4:

$$\hat{y} = r_1(s_j \pm 2^i) \prod_{k \in S - s_j} s_k \pmod N$$

If Bob knows that the error indeed occurred in s_j*, the following simple operations enable Bob to calculate* s_j*:*

$$C_1 = \frac{\hat{y}}{r_2^2} \pmod N = \quad (1)$$

$$= (s_j \pm 2^i)^2 \prod_{k \in S - s_k} s_k^2 \pmod N \quad (2)$$

$$C = \frac{C_1}{\prod_{i \in S} u_i} \pmod N = \quad (3)$$

$$= \frac{u_j + 2^{2i} \pm 2^{i+1} s_j}{u_j} \pmod N \quad (4)$$

$$s_j = \pm \frac{u_j(C-1) - 2^{2i}}{2^{i+1}} \pmod N \quad (5)$$

This calculation requires three multiplications in steps 2 and 4. Step 5 requires $\mathrm{O}(n)$ *trials (modular multiplications) to identify the correct* s_j*, by determining the correct error location* i*. Thus, the complexity to calculate* s_j *is* $\mathrm{O}(n)$*. Considering that Bob does not know a priori which* s_j *includes the error, he must repeat this procedure for all* $|S|$ *possible* s_j*'s to identify the correct one. Thus, the total complexity to derive one* s_j *is* $\mathrm{O}(n\ell)$*.*

Given ℓ *erroneous protocol invocations, so that errors occur in every* s_i*,* $i \in \{1, \ldots, \ell\}$*, Bob is able to derive all secret elements of Alice in the time required to perform* $\mathrm{O}(n\ell^2)$ *modular exponentiations.*

3.3. Practical considerations

For a successful impersonation, the attacker needs to extract all secret information from the device under attack. Using this information, the attacker can create a clone device of a legitimate one. The clone device is indistinguishable from the legitimate one, because it operates identically.

Once the attacker has a method to introduce transient bit flips during the execution of the cryptographic protocol, he can collect the erroneous output online (by interacting with the device) and then work offline (without interacting with the device) to extract the secret information. This approach minimizes the interaction time of the attacker with the device under attack; once he has collected all required erroneous output, the legitimate device is not required further.

The time required to extract the secret information is polynomial to both n and ℓ. Current implementations of the Fiat-Shamir scheme, especially in resource-limited devices, use small values for the parameter ℓ (e.g., $\ell = 7$ or $\ell = 9$) and $n = 512$ or $n = 1024$. For such values, the time required to extract the secret information is practically insignificant, given the enormous computing power available even in typical personal computers.

4. Conclusions

Fault-injection attacks constitute a realistic threat for systems that implement cryptographic protocols. We presented a fault-injection attack that can be used to extract the secret information from systems that implement either the classical Fiat-Shamir scheme or the *Precautious Fiat-Shamir Scheme*. The latter has been introduced as a countermeasure for the Bellcore attack. In this paper, we have described a fault model that has stronger assumptions than the one of the Bellcore attack and slightly higher complexity. However, it is successful against both schemes.

References

[1] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side-Channel(s). In B. S. K. Jr., Çetin Kaya Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems, CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 29–45. Springer-Verlag, 2002.

[2] R. Anderson and M. Kuhn. Tamper Resistance – a Cautionary Note. In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, November 1996.

[3] E. Antoniadis, D. Serpanos, A. Traganitis, and A. Voyiatzis. Software Simulation of Active Attacks on Cryptographic Systems. Technical Report TR-CSD-2001-01, Department of Computer Science, University of Crete, 2001.

[4] C. Aumüller, P. Bier, W. Fischer, P. Hofreiter, and J.-P. Seifert. Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures. In B. S. K. Jr., Çetin Kaya Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems, CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 260–275. Springer-Verlag, 2002.

[5] E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In *Advances in Cryptology-Crypto '97, LNCS 1294*, pages 513–525. Springer-Verlag, 1997.

[6] D. Boneh, R. DeMillo, and R. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *Advances in Cryptology, EUROCRYPT '97, LNCS 1233*, pages 37–51. Springer-Verlag, 1997.

[7] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of eliminating errors in cryptographic computations. *Journal of Cryptology*, 14(2):101–119, 2001.

[8] D. Brumley and D. Boneh. Remote Timing Attacks Are Practical. In *Proceedings of the 12th USENIX Security Symposium*, pages 1–14, August 2003.

[9] P. Dusart, G. Letourneux, and O. Vivolo. Differential Fault Analysis on A.E.S. In *Applied Cryptography and Network Security 2003, LNCS 2846*, pages 293–306. Springer-Verlag, 2003.

[10] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic Analysis: Concrete Results. In *Cryptographic Hardware and Embedded Systems, CHES 2001, LNCS 2162*, pages 251–261. Springer-Verlag, 2001.

[11] P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems. In *Crypto '96, LNCS 1109*, pages 104–113. Springer-Verlag, 1996.

[12] P. Kocher, J. Jaffe, and J. Benjamin. Differential Power Analysis. In *Crypto '99, LNCS 1666*, pages 388–397. Springer-Verlag, 1999.

[13] D. P. Macher. Fault Induction Attacks, Tamper Resistance, and Hostile Reverse Engineering in Perspective. In R. Hirschfeld, editor, *Financial Cryptography, FC '97 Proceedings*, volume 1318 of *Lecture Notes in Computer Science (LNCS)*, pages 109–121. Springer, 1997.

[14] A. Voyiatzis and D. Serpanos. Active Hardware Attacks and Proactive Countermeasures. In 7^{th} *IEEE Symposium on Computers and Communications (ISCC 2002)*, July 2002. ISBN: 0-7695-1671-8.