Das Dual-EC-DRBG Desaster

Sebastian Gottwald

Universität Leipzig

Zusammenfassung

Der Dual Elliptic Curve Pseudo Random Bit Generator (Dual-EC-DRBG) wurde zum Generieren von pseudo-zufälligen Zahlen auf Basis einer zufälligen Eingabe genutzt. Die daraus resultierenden Bitstreams bilden die Grundlage für viele kryptographische Algorithmen. Kurze Zeit nach der Veröffentlichung kamen Bedenken auf, dass der Dual-EC nicht sicher sei. Es wurde vermutet, die habe NSA eine Hintertür in den Dual-EC-DRBG eingebaut. Dennoch wurde das Verfahren vom NIST (National Institute of Standards and Technology), ANSI und ISO standardisiert und von einigen Krypto-Anwendungen genutzt.

1 Einführung

Der Dual-EC wurde Anfang der 2000er Jahre entwickelt und 2004 öffentlich bei einem NIST-Workshop präsentiert. Er gilt als äußerst langsam und schlecht designed. Dies wurde jedoch mit der Behauptung gerechtfertigt, dass der Dual-EC dafür eine höhere Sicherheit aufweise. Der Dual-EC wurde von E.Barker und J.Kelsey [1] Ende 2005 als NIST-Standard vorgeschlagen. Zuvor hatte die NSA bereits einen 10 Millionen Deal mit RSA gemacht und dafür gesorgt, dass der Dual-EC in die BSAFE Bibliothek eingebaut wird. Kurze Zeit nach der Präsentation des Algorithmus, wurde der Dual-EC wegen seiner Laufzeit und einer möglichen Hintertür kritisiert.

Mit den Enthüllungen von Edward Snowden wurde der Verdacht der Hintertür bestätigt und NIST entfernte den Pseudozufallszahlengenerator aus ihrem Standard.

Diese Arbeit bezieht sich auf die Veröffentlichungen von Schoenmaker und Sidorenko[4] [2], der Arbeit von Bernstein, Lange und Niederhagen [3] sowie dem Blog von Matthew Green [7].

2 Wie der Dual-EC-DBRG funktioniert

In diesem Kapitel wird die Funktionsweise des Dual-EC beschrieben. Es wurden 2 verschiedene Versionen für den NIST Standard vorgeschlagen. Da es bereits vor dem Standardisierungsprozess zu Sicherheitsbedenken kam wurden bei der 2006er Version zusätzliche zufällige Eingaben hinzugefügt, welche die Hintertür teilweise verschlossen. Diese wurde jedoch in der späteren 2007er Version repariert.

2.1 Allgemeiner Aufbau des PRNG

Im Folgenden wird der allgemeine Aufbau eines Pseudozufallszahlengenerators beschrieben. Dabei beschreibt Algorithmus 1 die Rechnungen als Pseudocode.

Dual-EC spezifiziert zwei Punkte P und Q mit $P = (x_p, y_p)$ und $Q = (x_q, y_q)$ auf der Standard NIST P-256 Elliptischen Kurve $E(\mathbb{F}_p)$. Eine Konstante α mit $P = \alpha Q$ ist somit aufgrund des *elliptic curve discrete logarithm problem* sehr schwer zu berechnen, selbst wenn P und Q bekannt sind.

Algorithm 1: Dual Elliptic Curve pseudorandom bit generator

Input: $s_0 \in \{0, 1, \ldots, \#E(\mathbb{F}_p) - 1\}$, $k > 0$
Output: 240k bits
for *i = 1 to k* **do**
  Set $s_i \leftarrow (s_{i-1}P)_x$ // Berechnung des neuen internen Zustand
  Set $r_i \leftarrow lsb_{240}((s_iQ)_x)$ //Berechnung der Ausgabe
end
Return $r_1, \ldots, r_k$

Der interne Startzustand s_0 ist ein zufälliger 256-bit Integer. Die Funktion

$$s_{i+1} = f(s_i) = (s_iP)_x$$

berechnet das s_i-vielfache von P und setzt den Wert des upgedateten internen Zustands als die x-Koordinate des resultierenden Punktes auf der Kurve. Die Funktion, welche die pseudo-zufällige Ausgabe generiert, ist mit

$$r_i = g(s_i) = lsb_{240}((s_iQ)_x)$$

beschrieben. Die Funktion $lsb_i(s)$ gibt die i-letzten bits einer Integer-Zahl aus. $lsb_3(23) = 7$ da $23 = (10111)_2$. Bei dem Standard-Dual-EC Verfahren werden die ersten 16-bit entfernt. Wenn eine Ausgabe aus mehr als 240-bit bestehen soll, so wird der nächste innere Zustand s_{i+1} berechnet und dessen Ausgabe r_{i+1} so lange angefügt, bis die benötigte Ausgabelänge erreicht ist. Abbildung 1 zeigt einen Pseudozufallszahlengenerator als Ablaufdiagramm.

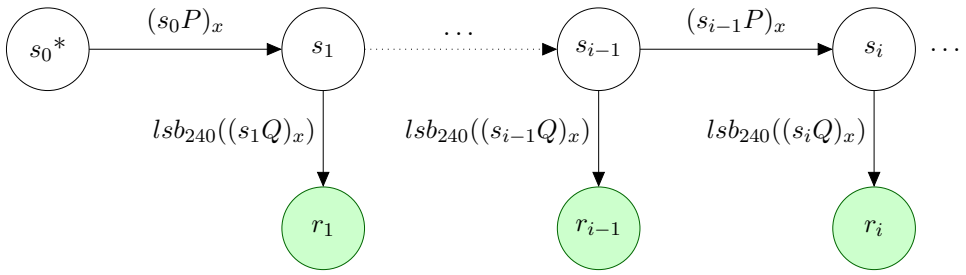

Abbildung 1: Grundlegender Aufbau des Pseudozufallszahlengenerators. P und Q sind Punkte auf einer elliptischen Kurve.

2.2 Dual-EC-2006

Bei der 2006er Version des Dual-EC gab es die Möglichkeit eine zusätzliche Eingabe zu generieren. Das kann z.B. die Systemzeit, die Temperatur oder eine andere zufällige Zahl sein.

Wenn der Dual-EC eine Ausgabe generieren soll, so wird die zusätzliche Eingabe $adin_i$ zur normalen Eingabe mit XOR zusammengefügt. Dadurch wird ein Zwischenzustand t_i errechnet, bevor der neue innere Zustand berechnet wird. Abbildung 2 zeigt das diesen Prozesses als Ablaufdiagramm. Wenn mehr als 240-bit benötigt werden, wird ohne vorher einen weiteren Zwischenzustand t zu berechnen, der nächste innere Zustand berechnet.

Diese Konstellation schließt die Hintertür zumindest für 30-byte Ausgaben. Bei längeren Ausgaben besteht die Hintertür allerdings weiterhin (siehe Abschnitt 3.3).

2.3 Dual-EC 2007

Da es Bedenken gab, dass die 2006er Version des Dual-EC es erlaube von einem internen Zustand vorangegangene Zustände und somit auch Ausgaben zu rekonstruieren, wurde eine weitere Version entwickelt. Da $s_{i+1} = (s_iP)_x$ bereits eine kryptografisch sichere One-Way Funktion ist, waren

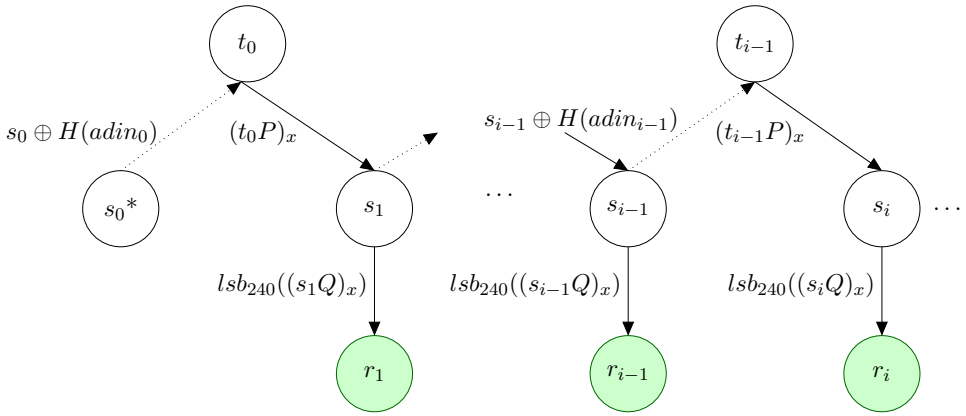

Abbildung 2: Aufbau der 2006er Version des Dual-EC. Zu dem internen Zustand wird eine zufällige Eingabe mittels XOR "addiert", bevor der nächste innere Zustand berechnet wird. Bei Zahlen länger als 240-bit wird kein temporärer Zustand t berechnet.

diese Bedenken jedoch unberechtigt. Abbildung 3 zeigt den Aufbau dieses modifizierten Dual-EC. Dual-EC 2007 verlangt einen zusätzlichen Update-Schritt nach jedem Aufruf. Das heißt, nachdem eine pseudozufällige Ausgabe generiert wurde, wird der interne Zustand ein weiteres Mal upgedatet $s_{i+1} = (s_iP)_x$. Aufgrund dieses zusätzlichen Updates kann der Angreifer die Attacke wieder ausführen.

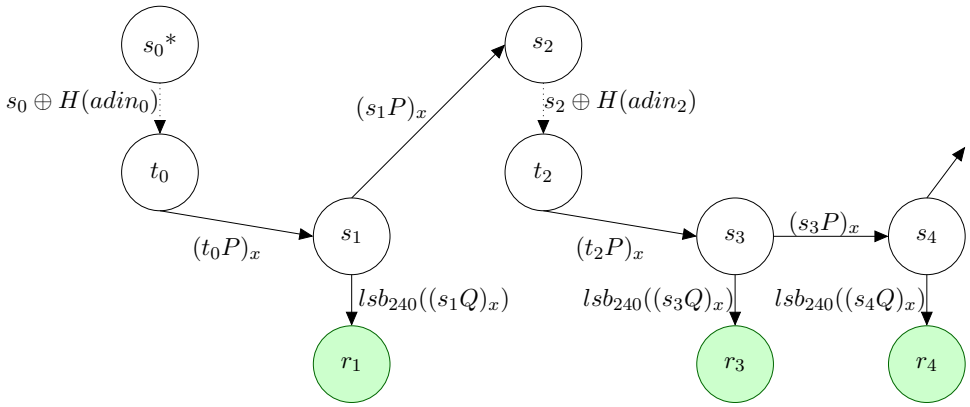

Abbildung 3: Der Dual-EC in der 2007er Version. Der innere Zustand wird nach einer Ausgebe ein weiteres Mal geupdated, bevor ein Zwischenzustand errechnet wird. r_2 und r_3 bilden eine Ausgabe, die länger als 240-bit ist.

3 Die Sicherheitsprobleme des Dual-EC

Der Dual-EC besitzt mehrere Sicherheitsprobleme, welche im Folgenden erläutert werden.

3.1 Der fehlende Sicherheitsbeweis

Als der Dual-EC von Barker und Kelsey für den NIST-Standard vorgeschlagen wurde, behaupteten diese, der Dual-EC-DRGB würde sich auf das elliptic curve discrete logarithm problem (ECDLP) reduzieren lassen. Dieses Problem gilt als schwer zu lösen und Algorithmen die sich darauf reduzieren lassen, werden als sicher angesehen. Sie führten jedoch keine Reduktion auf dieses Problem durch. Trotz eines fehlenden Sicherheitsbeweises wurde der Dual-EC in den Standard aufgenommen.

3.2 Die Ausgabe von Bits

Die Berechnung der Zustandsfolge s_iQ, lässt sich nach Brown nicht von einer von einer uniform zufälligen Auswahl von Punkten auf der elliptischen Kurve unterscheiden, solange das *DDH Problem*(decisional Diffie-Hellman Problem), das ECDLP (elliptic curve deterministic logarithm Problem) sowie das truncated point Problem gelöst werden kann [8]. Die x-Koordinaten der Punkte s_iQ lassen sich jedoch von uniform zufälligen gewählten Zahlen *mod p* unterscheiden. Grund hierfür ist, dass nur ca. die Hälfte der Punkte in $0 \leq x \leq p-1$ x-Koordinaten der Kurve sind. Beim Dual-EC wird versucht, dieses Problem mit der $lsb_i(s)$ Funktion zu beheben.

Die Ausgabe lässt sich jedoch nach Schoemakers und Sidorenko trotzdem von einer uniform zufälligen Ausgabe unterscheiden [4].

Schoemakers und Sidorenko lieferten 2006 einen Algorithmus, der mit erhöhter Wahrscheinlichkeit die Ausgabe des Dual-EC von einer echt zufälligen Ausgabe eines Zahlengenerators unterscheiden kann. Sie erstellten einen Unterscheider der mit einer Wahrscheinlichkeit von rund 50.156% zu errät, ob eine 240-bit Nachricht von dem Dual-EC stammt oder nicht. Bei einer längeren Ausgabe von 4000 240-bit Blöcken betrug die Wahrscheinlichkeit des richtig Ratens sogar rund 59.757%.

Ein weiteres Problem der lsb_{240} Funktion ist, dass es zu einfach ist die x-Koordinate zu errechnen, da lediglich die fehlenden 16 bit erraten werden müssen. Diese 2^{16} Rechenoperationen kann jeder Computer schnell ausführen. Ein Angreifer müsste dann jedoch noch das elliptic curve discrete logarithm problem lösen, um den Dual-EC zu brechen.

3.3 Die Hintertür

Damit der Dual-EC sicher ist, darf es für einen Angreifer keine Möglichkeit geben, einen inneren Zustand s_i zu errechnen. Dadurch könnten alle Folgezustände und somit auch alle Ausgaben reproduziert werden. Die Attacke, welche von Shumow und Ferguson [5] vorgeschlagen wurde, kann wie folgt beschrieben werden:

Angenommen der Angreifer kennt eine Konstante d, sodass $P = dQ$. Es lässt sich zeigen, dass das Ermitteln des internen Zustandes s_i recht einfach ist.

Der Angreifer sieht die pseudo-zufällig generierte Ausgabe r_1 (r_1 wird z.B. als öffentlicher Nonce [Zahl die nur einmal verwendet wird, z.B. beim Zurücksetzten eines Passwortes] benutzt). So kann der Angreifer mit Hilfe der Kurvenfunktion eine y-Koordinate berechnen, die zur x-Koordinate von r_1 gehört. Die Entfernung der ersten 16-bit mit der Funktion $lsb_240(x)$ kann vernachlässigt werden, da der Angreifer alle 2^{16} Möglichkeiten ausprobieren kann.

Er erhält eine Koordinate $R = (r_1, y_{r1}) = s_1Q$. Die Variable s_1 ist der interne Zustand und für den Angreifer unbekannt. Danach berechnet er $d \cdot R = d \cdot s_1Q = s_1dQ = s_1P$ und erhält somit den internen Zustand s_2 und alle möglichen Folgezustände, da $s_2 = (s_1P)_x$.

Das heißt, wenn es möglich ist die Konstante d herauszufinden, so ist der Dual-EC gebrochen. Das Problem ist, dass das Errechnen von d nach dem *elliptic curve discrete logarithm problem* sehr komplex ist. Beim Dual-EC wurde jedoch nicht bekannt gegeben, wie die Punkte P und Q generiert wurden. Es besteht also die Möglichkeit, dass der Entwickler des Dual-EC P und Q mit Hilfe von d generiert hat und somit die Hintertür eingebaut hat.

3.3.1 Die Hintertür bei der 2006er Version

Bei der 2006er Version wurde die Hintertür zumindest für 30 Byte Ausgaben verschlossen. Zur Erinnerung: Der Dual-EC berechnet nach jeder Ausgabe einen temporären Zwischenzustand mit Hilfe einer zusätzlichen Eingabe. Lediglich bei Ausgaben über 240 bit wird der temporäre Zustand nicht berechnet, sondern der Dual-EC geht direkt in den inneren Zustand. Bei einer 30 Byte Ausgabe ergibt sich Folgendes: Angenommen der Angreifer sieht r_1, so kann er s_1 nicht berechnen, da s_1 mit der zusätzlichen Eingabe modifiziert wurde und somit vom Zwischenzustand t_0 abhängt. Selbst wenn der Angreifer $adin_1$ errät, so kann er den inneren Zustand von r_1 nicht berechnen.

3.3.2 Die Hintertür bei der 2007er Version

Bei der 2007er Version des Dual-EC wird die Hintertür wieder repariert. Hier wird, nachdem eine Ausgabe r_i gemacht wurde, zunächst der innere Zustand upgedatet, bevor mittels einer zufälligen Eingabe ein weiterer Zwischenzustand berechnet wird.

Bei einer zufälligen Ausgabe r_1, kann der Angreifer einen Punkt $R = (r_1, x_{r1})$ und somit $s_2 = (d_{s_1}Q)_x = (s_1P)_x$ errechnen. Der Angreifer muss um r_2 zu berechnen noch die zusätzliche Eingabe erraten, vorausgesetzt das Opfer verwendet diese. Diese Änderung öffnet die Hintertür für den Angreifer wieder.

4 Zusammenfassung

Der Dual-EC DRBG Algorithmus war langsam und unsicher. Trotz der Sicherheitsbedenken wurde er als Standard aufgenommen und als Grundlage von etlichen Sicherheitsanwendungen verwendet.

Nach den Veröffentlichungen Edward Snowdens im Jahr 2013 riet NIST dazu, den Algorithmus nicht mehr zu verwenden und entfernte Dual-EC aus dem Standard.

Literatur

[1] BARKER, Elaine; KELSEY, John. *Recommendation for random number generation using deterministic random bit generators*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2006.

[2] FARASHAHI, Reza Rezaeian; SCHOENMAKERS, Berry; SIDORENKO, Andrey. *Efficient pseudorandom generators based on the DDH assumption*. In: International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, 2007. S. 426-441.

[3] BERNSTEIN, Daniel J.; LANGE, Tanja; NIEDERHAGEN, Ruben. *Dual EC: a standardized back door*. In: The New Codebreakers. Springer Berlin Heidelberg, 2016. S. 256-281.

[4] SCHOENMAKERS, Berry; SIDORENKO, Andrey. Cryptanalysis of the Dual Elliptic Curve Pseudorandom Generator. IACR Cryptology ePrint Archive, 2006, 2006. Jg., S. 190.

[5] SHUMOW, Dan; FERGUSON, Niels. On the possibility of a back door in the NIST SP800-90 Dual EC Prng. In: Proc. Crypto. 2007.

[6] CHECKOWAY, Stephen, et al. On the Practical Exploitability of Dual EC DRBG in TLS Implementations. 2014.

[7] GREEN, Matthew; The Many Flaws of Dual_EC_DRBG; `https://blog.cryptographyengineering.com/2013/09/18/the-many-flaws-of-dualecdrbg/`; 12.06.17

[8] BROWN, Daniel RL. Conjectured Security of the ANSI-NIST Elliptic Curve RNG. IACR Cryptology ePrint Archive, 2006, 2006. Jg., S. 117.