

# Bitsicherheit des Diffie-Hellman Problems

## Sicherheitsergebnis

Sei  $p$  eine Primzahl mit  $n$  Bit und  $g \in \mathbb{F}_p$  ein Erzeuger der multiplikativen Gruppe von  $\mathbb{F}_p$ . Das Diffie-Hellman Problem (DHP oder CDH) ist, aus  $g, g^a, g^b$  den Wert  $g^{ab}$  zu bestimmen. Die obersten Bits von  $g^{ab}$  sind so schwer zu berechnen wie alle Bits:

**1 Satz (Boneh-Venkatesan).** *Sei  $\varepsilon > 0$  eine absolute Konstante und  $\ell = \lceil \varepsilon n^{1/2} \rceil$ . Sei  $A$  ein Algorithmus, der nach Eingabe von  $g, g^a, g^b$  die obersten  $\ell$  Bits von  $g^{ab}$  in erwarteter Polynomzeit in  $n$  berechnet. Dann gibt es einen Algorithmus  $B$ , der nach Eingabe von  $g, g^a, g^b$  und der Faktorisierung von  $p-1$  unter Verwendung von  $B$  den Wert  $g^{ab}$  in erwarteter Polynomzeit in  $n$  berechnet.*

Der Satz wird auf das Hidden Number Problem (HNP) wie folgt zurückgeführt.

Als Vorberechnung wählen wir der Reihe nach zufällige  $r$ , bis  $g^{a+r}$  ein Erzeuger von  $\mathbb{F}_p^\times$  ist. Dies können wir mit der Faktorisierung von  $p-1$  leicht testen. Die relative Anzahl der Erzeuger ist  $\phi(p-1)/(p-1) = \Omega(1/\log(\log(p)))$ , so daß  $O(\log(\log(p)))$  Versuche ausreichen sollten. In praktischen Anwendung ist die Faktorisierung von  $p-1$  a priori bekannt, da man  $p$  gerade so konstruiert, daß  $p-1$  nur wenige kleine Primfaktoren besitzt. Entsprechend ist auch die Wahrscheinlichkeit recht hoch, daß  $g^{a+r}$  für zufälliges  $r$  ein Erzeuger ist.

Nach der Vorberechnung wenden wir  $A$  auf die Elemente  $g^{a+r}$  und  $g^{b+t}$  für viele zufällige Werte  $t$  an. Dies ergibt die obersten Bits von  $g^{(a+r)b} g^{(a+r)t}$ , wobei die  $g^{(a+r)t}$  gleichverteilt zufällige und unabhängige Elemente aus  $\mathbb{F}_p^\times$  sind und  $g^{ab}$  leicht aus den bekannten Größen  $r, g^b$  und  $g^{(a+r)b}$  berechnet werden kann.

Dies führt auf das Hidden Number Problem. Mit  $\text{MSB}_\ell(x)$  seien die obersten  $\ell$  Bits von  $x \in \mathbb{Z}$  mit  $0 \leq x \leq p-1$  bezeichnet, also eine Zahl  $z$  mit

$0 \leq z \leq p-1$  und  $0 \leq x-z < p/2^\ell$ . Für  $x \in \mathbb{Z}$  bezeichnen wir mit  $x \bmod p$  den Vertreter von  $x + \mathbb{Z}p$  in  $[0, p-1]$ . Gegeben sind gleichverteilt zufällige und unabhängige  $t_1, \dots, t_d \in \mathbb{Z}$  mit  $1 \leq t_i \leq p-1$  und  $a_i = \text{MSB}_\ell(\alpha t_i \bmod p)$  für  $1 \leq i \leq d$  und für ein unbekanntes Element  $\alpha \in \mathbb{Z}$ ,  $1 \leq \alpha \leq p-1$ . Das HNP ist, aus diesen Daten  $\alpha$  zu berechnen.

## Gitterangriff auf das Hidden Number Problem

Wir verallgemeinern das HNP in der folgenden Form. Mit  $p$  wird wieder eine große Primzahl bezeichnet. Ferner seien  $d \in \mathbb{Z}^{\geq 1}$  und  $\alpha \in \mathbb{Z}$  mit  $1 \leq \alpha \leq p-1$  gegeben. Für  $k \in \mathbb{R}$  mit  $1 \leq k \leq \log_2(p)$  betrachten wir Funktionen  $f_k : \mathbb{Z} \rightarrow \mathbb{Z}$  mit  $|x - f_k(x)| \leq p2^{-k}$  für alle  $x \in \mathbb{Z}$ . Gegeben seien weiter gleichverteilt zufällige und unabhängige gewählte  $t_1, \dots, t_d \in \mathbb{Z}$  mit  $1 \leq t_i \leq p-1$ . Die Aufgabe besteht darin, aus  $p, t_1, \dots, t_d$  und den Werten  $f_k(\alpha t_1), \dots, f_k(\alpha t_d)$  das versteckte Element  $\alpha$  zu berechnen.

Das HNP kann auf ein CVP bezüglich des folgenden Gitters zurückgeführt werden. Wir betrachten das von den Zeilen der  $(d+1) \times (d+1)$  Matrix

$$\begin{pmatrix} pI_d & 0 \\ t_1 \cdots t_d & 1/p \end{pmatrix}$$

erzeugte, vollständige Gitter  $\Lambda$ . Der Vektor  $v = (\alpha t_1 \bmod p, \dots, \alpha t_d \bmod p, \alpha/p)$  liegt in  $\Lambda$ , denn er ist die letzte Zeile multipliziert mit  $\alpha$  plus eine Linearkombination der vorhergehenden Zeilen. Ist  $d$  klein gegenüber  $p$ , so ist  $a = (a_1, \dots, a_d, 0)$  mit  $a_i = f_k(\alpha t_i)$  ein Vektor in  $\mathbb{R}\Lambda$ , welcher nahe an  $\Lambda$  liegt, weil  $\|v - a\| \leq (d+1)^{1/2}p2^{-k}$  gilt. Im Umkehrschluß wollen wir ein CVP bezüglich  $\Lambda$  und  $a$  lösen, und hoffen, daraus  $v$  beziehungsweise  $\alpha$  zu erhalten. Hierzu sind zwei Fragen zu beantworten:

1. Mit welchem Approximationsfaktor können wir das CVP überhaupt berechnen?
2. Inwiefern bestimmt der von uns berechnete, nahe Gittervektor  $w$  überhaupt  $v$  beziehungsweise  $\alpha$ ?

Die erste Frage soll auf später verschoben werden. Wir gehen hier davon aus, daß wir allgemein gesprochen das CVP für ein Gitter  $\Lambda$  der Dimension  $d$  und den Vektor  $a \in \mathbb{R}\Lambda$  mit einem Vektor  $w \in \Lambda$  der Form  $\|w - a\| \leq 2^{d/2}\|v - a\|$  für alle  $v \in \Lambda$  effizient lösen können. Es ist sogar möglich, zu festem  $\varepsilon > 0$  das CVP in der Form  $\|w - a\| \leq 2^{\varepsilon d}\|v - a\|$  effizient zu lösen. Die Abhängigkeit der Laufzeit von  $1/\varepsilon$  ist dabei natürlich exponentiell.

Wir wenden uns daher der zweiten Frage zu.

**2 Lemma.** Sei  $\mu \in \mathbb{R}^{\geq 2}$ . Mit den obigen Definitionen seien die  $t_i$  zufällig gleichverteilt und unabhängig gewählt. Für jedes  $b = (b_1, \dots, b_d, 0) \in \mathbb{R}^{d+1}$  mit  $\|v - b\| \leq p2^{-\mu}$  gilt

$$\Pr \left( \begin{array}{l} \forall w \in \Lambda : \\ \|w - b\| \leq p2^{-\mu} \Rightarrow \end{array} \begin{array}{l} w = (\alpha t_1 \bmod p, \dots, \alpha t_d \bmod p, \beta/p) \\ \text{für ein } \beta \in \mathbb{Z} \text{ mit } \beta \equiv \alpha \bmod p \end{array} \right) \geq 1 - p2^{-(\mu-3)d}.$$

*Beweis.* Die Wahrscheinlichkeit beschreibt den Teil der möglichen Gitter  $\Lambda$ , für welche die Implikation gilt.

Jedes  $w \in \Lambda$  ist von der Form  $w = (\beta t_1 - z_1 p, \dots, \beta t_d - z_d p, \beta/p)$ . Wir unterscheiden jetzt Fälle, je nachdem, welchen Wert  $\beta$  annimmt.

Für  $\beta \equiv \alpha \bmod p$  folgt  $\beta t_i - z_i p = \alpha t_i \bmod p$ , da sonst  $\|w - b\| \geq |w_i - b_i| \geq |w_i - v_i| - |v_i - b_i| \geq p - p2^{-\mu} > p2^{-\mu}$  unter Beachtung von  $w_i \equiv v_i \bmod p$  und  $|v_i - b_i| \leq \|v - b\| \leq p2^{-\mu}$  gilt.

Für  $\beta \not\equiv \alpha \bmod p$  definieren wir  $\text{dist}_p(x, y) = \min\{|x - y - \lambda p| \mid \lambda \in \mathbb{Z}\}$ . Es gilt  $\text{dist}_p(x, z) \leq \text{dist}_p(x, y) + \text{dist}_p(y, z)$ . Für zufälliges und gleichverteilt gewähltes  $t$  haben wir  $\Pr(\text{dist}_p(\alpha t, \beta t) > s) \geq 1 - (2s+1)/p$  (man kann hierfür zum Beispiel  $\alpha t$  als Ursprung auffassen und schätzt ab, wie der Zufallswert  $\beta t$  im Intervall der Länge  $2s$  um  $\alpha t$  liegt, wobei sich  $\alpha t$  auf einem Kreisumfang der Länge  $p$  befindet). Für  $s = 2p2^{-\mu}$  gilt

$$\Pr(\text{dist}_p(\alpha t_i, \beta t_i) > 2p2^{-\mu}) \geq 1 - ((4p2^{-\mu}) + 1)/p \geq 1 - 5 \cdot 2^{-\mu}$$

für jedes  $1 \leq i \leq d$ . Weiter ergibt sich für ein festes  $\beta$

$$\Pr(\exists i : \text{dist}_p(\alpha t_i, \beta t_i) > 2p2^{-\mu}) \geq 1 - (5 \cdot 2^{-\mu})^d$$

und für variables  $\beta$  wegen der  $p - 1$  Möglichkeiten für  $\beta$

$$\Pr(\exists \beta \not\equiv \alpha \bmod p, \exists i : \text{dist}_p(\alpha t_i, \beta t_i) > 2p2^{-\mu}) \geq 1 - (p - 1)(5 \cdot 2^{-\mu})^d.$$

Tritt das Ereignis dieser letzten Wahrscheinlichkeit ein, so gilt auch  $\|w - b\| \geq \|w - v\| - \|v - b\| > 2p2^{-\mu} - p2^{-\mu} = p2^{-\mu}$  im Widerspruch zur Annahme. Daher gilt in diesen Fällen  $\beta \equiv \alpha \bmod p$  und die Aussage über die Form von  $w$  ist korrekt. Für die Wahrscheinlichkeit aus dem Lemma gilt daher  $\geq 1 - (p - 1)(5 \cdot 2^{-\mu})^d \geq 1 - p2^{-(\mu-3)d}$ .  $\square$

**3 Lemma.** Sei  $k = \mu + (1/2) \log_2(d + 1) + (d + 1)/2 \leq \log_2(p)$ . Dann gibt es einen deterministischen polynomiellen Algorithmus  $A$ , so daß für jedes  $\alpha$  und zufällig gleichverteilt und unabhängig gewählte  $t_1, \dots, t_d$

$$\Pr(A(p, t_1, \dots, t_d, f_k(\alpha t_1), \dots, f_k(\alpha t_d)) = \alpha) \geq 1 - p2^{-(\mu-3)d}$$

gilt.

*Beweis.* Übungsaufgabe. □

Wenn wir bei konstanter Erfolgswahrscheinlichkeit  $k$  minimieren wollen, so erhalten wir das folgende. Für konstante Erfolgswahrscheinlichkeit benötigen wir ungefähr  $\mu d \approx \log_2(p)$  (anschaulich ergibt jeder Wert  $f_k(\alpha t_i)$   $k$  Bits von  $\alpha$ ). Um dann  $k$  zu minimieren, ist größenordnungsmäßig  $\mu \approx d \approx (\log_2(p))^{1/2}$  erforderlich, folglich  $k \approx (\log_2(p))^{1/2}$ .

Mit Hilfe der verbesserten CVP Approximation kann man  $k$  weiter optimieren. Die Form von  $k$  ist dann ungefähr  $k \approx \mu + \varepsilon d$  mit  $\mu \approx (1/c)(\log_2(p))^{1/2}$ ,  $d \approx c(\log_2(p))^{1/2}$  und einem zu bestimmenden  $c$ . Einsetzen liefert  $k \approx (1/c + \varepsilon c)(\log_2(p))^{1/2}$ , welches für  $c \approx 1/\varepsilon^{1/2}$  minimal wird, also  $k \approx (\varepsilon \log_2(p))^{1/2}$ . Dies ergibt im wesentlichen die eingangs angegebene Aussage über die Bit sicherheit von DHP.

## Bemerkungen

Weitere Varianten des HNP werden in der Literatur betrachtet.

- Man kann zum Beispiel die Funktionen  $f_k$  durch irgendeine Auswahl von  $k$  Bits aus den  $\alpha t_i \bmod p$  geeignet ersetzen. Das Verfahren wird langsamer, wenn kleinere und unzusammenhängende Bitblöcke betrachtet werden.
- Man kann statt der Faktorisierung und der Forderung, daß  $g^{a+r}$  ein Erzeuger sein soll, abgeschwächte Forderungen an die Mindestgröße der Ordnung von  $g^{a+r}$  stellen. Die Faktorisierung ist dann nicht mehr erforderlich.
- Man kann die  $t_i$  bezüglich gewisser Verteilungen wählen (und nicht nur gleichverteilt und unabhängig).