A sceptical view on decomposition attacks

Claus Diem

University of Leipzig

June 6, 2016

Decomposition attacks

Fix some kind of finite abelian groups $(G, +)$.

Decomposition attacks

Fix some kind of finite abelian groups $(G, +)$.

Consider the **discrete logarithm problem**:

Given $(G, +)$ and A, B with $B = eA$ for some $e \in \mathbb{N}$

Decomposition attacks

Fix some kind of finite abelian groups $(G, +)$.

Consider the **discrete logarithm problem**:

Given $(G, +)$ and A, B with $B = eA$ for some $e \in \mathbb{N}$ compute such an e!

Decomposition attacks

Input. $(G, +); A, B \in G$, A a generator

Decomposition attacks

Input. $(G, +); A, B \in G$, A a generator

1. Compute $N := \mathrm{ord}(G)$.

Decomposition attacks

Input. $(G, +)$; $A, B \in G$, A a generator

1. Compute $N := \mathrm{ord}(G)$.
2. Fix a **decomposition base / factor base** $\mathcal{F} = \{F_1, \ldots, F_r\}$.

Decomposition attacks

Input. $(G, +)$; $A, B \in G$, A a generator

1. Compute $N := \mathrm{ord}(G)$.
2. Fix a **decomposition base / factor base** $\mathcal{F} = \{F_1, \ldots, F_r\}$.
3. Compute $r + 1$ **relations**

$$\alpha_i A + \beta_i B = \sum_j r_{i,j} F_j \ .$$

Decomposition attacks

Input. $(G, +)$; $A, B \in G$, A a generator

1. Compute $N := \mathrm{ord}(G)$.
2. Fix a **decomposition base / factor base** $\mathcal{F} = \{F_1, \ldots, F_r\}$.
3. Compute $r + 1$ **relations**

$$\alpha_i A + \beta_i B = \sum_j r_{i,j} F_j \ .$$

4. Compute $\underline{\gamma} \in \mathbb{Z}/N\mathbb{Z}$ with $\underline{\gamma} R = \underline{0}$.

Decomposition attacks

Input. $(G, +)$; $A, B \in G$, A a generator

1. Compute $N := \mathrm{ord}(G)$.
2. Fix a **decomposition base / factor base** $\mathcal{F} = \{F_1, \ldots, F_r\}$.
3. Compute $r + 1$ **relations**

$$\alpha_i A + \beta_i B = \sum_j r_{i,j} F_j \ .$$

4. Compute $\underline{\gamma} \in \mathbb{Z}/N\mathbb{Z}$ with $\underline{\gamma} R = \underline{0}$.

$$\Longrightarrow (\sum_i \gamma_i \alpha_i) A + (\sum_i \gamma_i \beta_i) B = 0$$

Decomposition attacks

Input. $(G, +)$; $A, B \in G$, A a generator

1. Compute $N := \mathrm{ord}(G)$.
2. Fix a **decomposition base / factor base** $\mathcal{F} = \{F_1, \ldots, F_r\}$.
3. Compute $r + 1$ **relations**

$$\alpha_i A + \beta_i B = \sum_j r_{i,j} F_j \ .$$

4. Compute $\underline{\gamma} \in \mathbb{Z}/N\mathbb{Z}$ with $\underline{\gamma} R = \underline{0}$.

$$\Longrightarrow (\sum_i \gamma_i \alpha_i) A + (\sum_i \gamma_i \beta_i) B = 0$$

5. Output $e := -\frac{\sum_i \gamma_i \beta_i}{\sum_i \gamma_i \alpha_i} \in \mathbb{Z}/N\mathbb{Z}$.

Decomposition attacks

Task. Compute **decompositions**:

Decomposition attacks

Task. Compute **decompositions**:

$$R$$

Decomposition attacks

Task. Compute **decompositions**:

$$R =$$

Decomposition attacks

Task. Compute **decompositions**:

$$R = r_1 F_1 + \cdots + r_r F_r \ .$$

Decomposition attacks

Task. Compute **decompositions**:

$$R = r_1 F_1 + \cdots + r_r F_r \ .$$

Or:

$$R$$

Decomposition attacks

Task. Compute **decompositions**:

$$R = r_1 F_1 + \cdots + r_r F_r \ .$$

Or:

$$R =$$

Decomposition attacks

Task. Compute **decompositions**:

$$R = r_1 F_1 + \cdots + r_r F_r \ .$$

Or:

$$R = P_1 + \cdots + P_m$$

with

$$P_1, \ldots, P_m \in \mathcal{F}$$

In elliptic curves

Given: $E/\mathbb{F}_{q^n}, n > 1$ with points A, B.

In elliptic curves

Given: $E/\mathbb{F}_{q^n}, n > 1$ with points A, B.

Recall: $x(P)$ determines $\pm P$.

In elliptic curves

Definition of $\mathcal{F}$:

Fix $U < \mathbb{F}_{q^n}$. Let

$$\mathcal{F} := \{P \in E(\mathbb{F}_{q^n}) \mid x(P) \in U\}.$$

In elliptic curves

Definition of $\mathcal{F}$:

Fix $U < \mathbb{F}_{q^n}$. Let

$$\mathcal{F} := \{P \in E(\mathbb{F}_{q^n}) \mid x(P) \in U\}.$$

Decomposition:

In elliptic curves

Definition of $\mathcal{F}$:

Fix $U < \mathbb{F}_{q^n}$. Let

$$\mathcal{F} := \{P \in E(\mathbb{F}_{q^n}) \mid x(P) \in U\}.$$

Decomposition:

$$R$$

In elliptic curves

Definition of $\mathcal{F}$:

Fix $U < \mathbb{F}_{q^n}$. Let

$$\mathcal{F} := \{P \in E(\mathbb{F}_{q^n}) \mid x(P) \in U\}.$$

Decomposition:

$$R = P_1 + \cdots + P_m$$

with $x(P_i) \in U,\ m \cdot \dim(U) \approx n$

In elliptic curves

Definition of $\mathcal{F}$:

Fix $U < \mathbb{F}_{q^n}$. Let

$$\mathcal{F} := \{P \in E(\mathbb{F}_{q^n}) \mid x(P) \in U\}.$$

Decomposition:

$$R = P_1 + \cdots + P_m$$

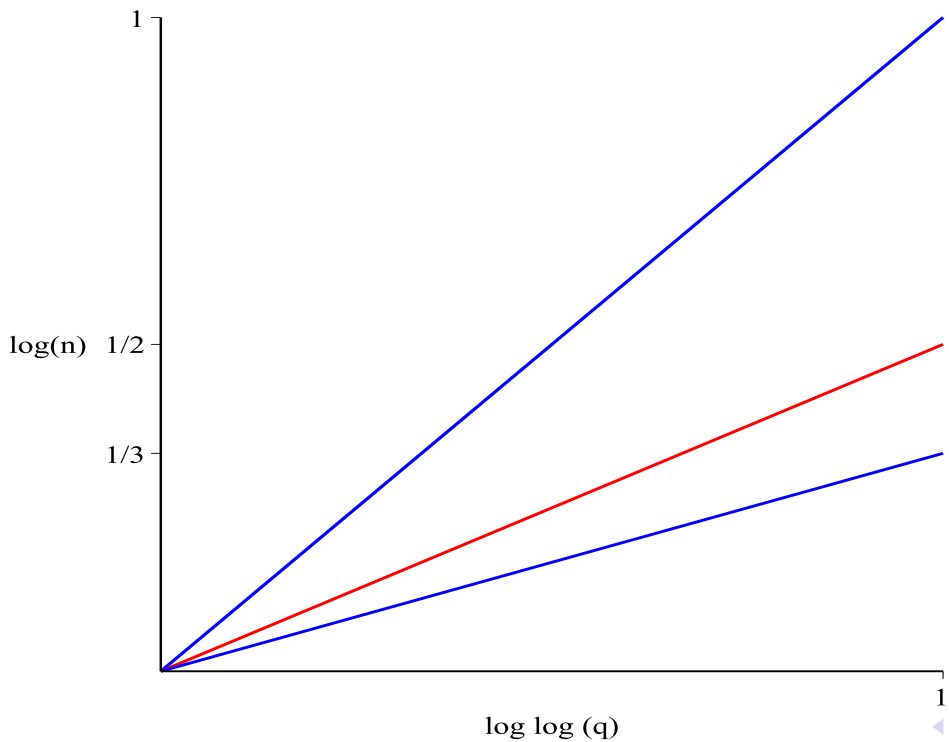
with $x(P_i) \in U,\ m \cdot \dim(U) \approx n$

via solving **polynomial systems** over $\mathbb{F}_q$.

A result

Theorem Let $a, b > 0$ be fixed. Then the DLP in elliptic curves over $\mathbb{F}_{q^n}$ can be solved in:

- For $a \cdot \log(q)^{1/2} \leq n \leq b \cdot \log(q)^{1/2}$: $\exp(O(\log(q^n)^{2/3}))$.
- For $a \cdot \log(q)^{1/3} \leq n \leq b \cdot \log(q)$: $\exp(O(\log(q^n)^{3/4}))$.



Solving polynomial systems

Let $F_1, \ldots, F_n \in k[X_0, \ldots, X_n]$ be homogeneous polynomials.

Solving polynomial systems

Let $F_1, \ldots, F_n \in k[X_0, \ldots, X_n]$ be homogeneous polynomials.

Then "generically", the system

$$F_1 = \cdots = F_n = 0$$

has exactly $B := \deg(F_1) \cdots \deg(F_n)$ solutions over $\overline{k}$.

Solving polynomial systems

Let $F_1, \ldots, F_n \in k[X_0, \ldots, X_n]$ be homogeneous polynomials.

Then "generically", the system

$$F_1 = \cdots = F_n = 0$$

has exactly $B := \deg(F_1) \cdots \deg(F_n)$ solutions over $\overline{k}$.

For $k = \mathbb{F}_q$ one can find all these solutions in a time of $(B \cdot \log(q))^{O(1)}$.

Solving polynomial systems

Let $F_1, \ldots, F_n \in k[X_0, \ldots, X_n]$ be homogeneous polynomials.

Then "generically", the system

$$F_1 = \cdots = F_n = 0$$

has exactly $B := \deg(F_1) \cdots \deg(F_n)$ solutions over $\overline{k}$.

For $k = \mathbb{F}_q$ one can find all these solutions in a time of $(B \cdot \log(q))^{O(1)}$.

1. Linear algebra on a system $X_i^d \cdot F_j \quad$ (variable i, j).

Solving polynomial systems

Let $F_1, \ldots, F_n \in k[X_0, \ldots, X_n]$ be homogeneous polynomials.

Then "generically", the system

$$F_1 = \cdots = F_n = 0$$

has exactly $B := \deg(F_1) \cdots \deg(F_n)$ solutions over $\overline{k}$.

For $k = \mathbb{F}_q$ one can find all these solutions in a time of $(B \cdot \log(q))^{O(1)}$.

1. Linear algebra on a system $X_i^d \cdot F_j \quad$ (variable i, j).
2. Factorization of a single univariate polynomial.

Solving polynomial systems

What if I have more equations then unknowns?

Solving polynomial systems

What if I have more equations then unknowns?

The output is now very small.

Solving polynomial systems

What if I have more equations then unknowns?

The output is now very small.

The running time is usually smaller, but ...

Solving polynomial systems

What if I have a system of **inhomogeneous** polynomials $f_1, \ldots, f_r \in \mathbb{F}_q[x_1, \ldots, x_n]$?

Solving polynomial systems

What if I have a system of **inhomogeneous** polynomials $f_1, \ldots, f_r \in \mathbb{F}_q[x_1, \ldots, x_n]$?

"Most of the time" it is like considering the system of leading terms.

Solving polynomial systems

What if I am just interested in solutions over $\mathbb{F}_q$?

Solving polynomial systems

What if I am just interested in solutions over $\mathbb{F}_q$?

One can add the **field equations**

$$x_i^q - x_i$$

Solving polynomial systems

What if I am just interested in solutions over $\mathbb{F}_q$?

One can add the **field equations**

$$x_i^q - x_i \ :$$

- irrelevant if degree q is not reached ("**large characteristic**")

Solving polynomial systems

What if I am just interested in solutions over $\mathbb{F}_q$?

One can add the **field equations**

$$x_i^q - x_i \ :$$

- irrelevant if degree q is not reached ("**large characteristic**")
- particularly relevant for $q = 2$.

Solving polynomial systems

What if I just want to find a **single** solution over $\mathbb{F}_q$?

Solving polynomial systems

What if I just want to find a **single** solution over $\mathbb{F}_q$?

Or whether there **exists** a solution over $\mathbb{F}_q$?

Solving polynomial systems

What if I just want to find a **single** solution over $\mathbb{F}_q$?

Or whether there **exists** a solution over $\mathbb{F}_q$?

This (usually) cannot be done faster.

Solving polynomial systems

What if I just want to find a **single** solution over $\mathbb{F}_q$?

Or whether there **exists** a solution over $\mathbb{F}_q$?

This (usually) cannot be done faster.

Compare: Over $\mathbb{R}$ one can efficiently approximate a single solution.

Large characteristic

Let $E/\mathbb{F}_{q^n}$ be given.

Consider $m = n$, $U = \langle v \rangle$.

We search for decompositions

$$R = P_1 + \cdots + P_n \ , \ x(P_i) \in U \ .$$

Large characteristic

Let $E/\mathbb{F}_{q^n}$ be given.

Consider $m = n$, $U = \langle v \rangle$.

We search for decompositions

$$R = P_1 + \cdots + P_n \ , \ x(P_i) \in U \ .$$

For example: Solve a system with n equations of degree 2^{n-1}.

Large characteristic

Let $E/\mathbb{F}_{q^n}$ be given.

Consider $m = n$, $U = \langle v \rangle$.

We search for decompositions

$$R = P_1 + \cdots + P_n \ , \ x(P_i) \in U \ .$$

For example: Solve a system with n equations of degree 2^{n-1}.

Heuristic: There are $2^{n(n-1)}$ solutions over $\overline{\mathbb{F}}_q$.

Large characteristic

Let $E/\mathbb{F}_{q^n}$ be given.

Consider $m = n$, $U = \langle v \rangle$.

We search for decompositions

$$R = P_1 + \cdots + P_n \ , \ x(P_i) \in U \ .$$

For example: Solve a system with n equations of degree 2^{n-1}.

Heuristic: There are $2^{n(n-1)}$ solutions over $\overline{\mathbb{F}}_q$.

Running time: $2^{\Theta(n^2)} \cdot \log^{\Theta(1)}(q)$.

Large characteristic

Let $E/\mathbb{F}_{q^n}$ be given.

Consider $m = n$, $U = \langle v \rangle$.

We search for decompositions

$$R = P_1 + \cdots + P_n \ , \ x(P_i) \in U \ .$$

For example: Solve a system with n equations of degree 2^{n-1}.

Heuristic: There are $2^{n(n-1)}$ solutions over $\overline{\mathbb{F}}_q$.

Running time: $2^{\Theta(n^2)} \cdot \log^{\Theta(1)}(q)$.

Faster?

Characteristic 2

Consider the DLP for $E/\mathbb{F}_{2^n}$.

Igor Semeav: Decompositions can be computed in **polynomial time**.

Characteristic 2

Consider the DLP for $E/\mathbb{F}_{2^n}$.

Igor Semeav: Decompositions can be computed in **polynomial time**.

Then the running time crucially depends on the probability that an R has a decomposition $R = P_1 + \cdots + P_m$

Characteristic 2

Consider the DLP for $E/\mathbb{F}_{2^n}$.

Igor Semeav: Decompositions can be computed in **polynomial time**.

Then the running time crucially depends on the probability that an R has a decomposition $R = P_1 + \cdots + P_m$ ...

... which should be about $\frac{1}{m!}$

Characteristic 2

Consider the DLP for $E/\mathbb{F}_{2^n}$.

Igor Semeav: Decompositions can be computed in **polynomial time**.

Then the running time crucially depends on the probability that an R has a decomposition $R = P_1 + \cdots + P_m$ ...

... which should be about $\frac{1}{m!}$ ...

... one obtains

$$e^{O(\sqrt{n \cdot \log(n)})}.$$

Characteristic 2

Consider the DLP for $E/\mathbb{F}_{2^n}$.

Igor Semeav: Decompositions can be computed in **polynomial time**.

Then the running time crucially depends on the probability that an R has a decomposition $R = P_1 + \cdots + P_m$ ...

... which should be about $\frac{1}{m!}$ ...

... one obtains

$$e^{O(\sqrt{n \cdot \log(n)})}.$$

I am going to argue:

If this is correct, then there is a **polynomial time** algorithm.

Increased probability

Traditional:

Fix $V < \mathbb{F}_{2^n}$.

Given R, compute $R = P_1 + \cdots + P_m$ with $x(P_i) \in V$.

Increased probability

Traditional:

Fix $V < \mathbb{F}_{2^n}$.

Given R, compute $R = P_1 + \cdots + P_m$ with $x(P_i) \in V$.

Variant:

Fix $\mathbb{F}_{2^n} = \bigoplus_{i=1}^{m} V_i$.

Given R, compute $R = P_1 + \cdots + P_m$ with $x(P_i) \in V_i$.

Increased probability

Traditional:

Fix $V < \mathbb{F}_{2^n}$.

Given R, compute $R = P_1 + \cdots + P_m$ with $x(P_i) \in V$.

Variant:

Fix $\mathbb{F}_{2^n} = \bigoplus_{i=1}^{m} V_i$.

Given R, compute $R = P_1 + \cdots + P_m$ with $x(P_i) \in V_i$.

Now set $m := n$, $V_i = \langle v_i \rangle$.

New algorithm

For points A, B,
find **one** decomposition / representation of the form

$$B = \pm A \pm 2A \cdots \pm 2^{n-3} A \, .$$

New algorithm

For points A, B,
find **one** decomposition / representation of the form

$$B = \pm A \pm 2A \cdots \pm 2^{n-3}A \, .$$

If it exists, it is unique.

New algorithm

For points A, B,
find **one** decomposition / representation of the form

$$B = \pm A \pm 2A \cdots \pm 2^{n-3}A \, .$$

If it exists, it is unique.

Equivalent:

$$B \pm A \pm 2A \pm \cdots \pm 2^{n-3}A = 0 \, .$$

New algorithm

For points A, B,
find **one** decomposition / representation of the form

$$B = \pm A \pm 2A \cdots \pm 2^{n-3}A \, .$$

If it exists, it is unique.

Equivalent:

$$B \pm A \pm 2A \pm \cdots \pm 2^{n-3}A = 0 \, .$$

$$S_3(x(B), x(A), x_1) = 0, \;\; S_3(x_1, x(2A), x_2) = 0, \ldots,$$
$$S_3(x_{n-4}, x(2^{n-2}A), x(2^{n-3}A)) = 0$$

Experimental results

Largest experiment for $n = 11$.

55 variables, 66 quadratic equations

Experimental results

Largest experiment for $n = 11$.

55 variables, 66 quadratic equations

Largest degree reached in Gröbner base computation:

Experimental results

Largest experiment for $n = 11$.

55 variables, 66 quadratic equations

Largest degree reached in Gröbner base computation: **3**

Experimental results

Largest experiment for $n = 11$.

55 variables, 66 quadratic equations

Largest degree reached in Gröbner base computation: **3**

... with 30 GB.

Experimental results

Largest experiment for $n = 11$.

55 variables, 66 quadratic equations

Largest degree reached in Gröbner base computation: **3**

... with 30 GB.

A heuristic says: For random systems usually:

Experimental results

Largest experiment for $n = 11$.

55 variables, 66 quadratic equations

Largest degree reached in Gröbner base computation: **3**

... with 30 GB.

A heuristic says: For random systems usually: 8

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity

$$n^2$$

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity

$$(n^2)^3$$

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5}$$

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage:

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage: $((n^2)^3)^2$ bits

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage: $((n^2)^3)^2$ bits $= n^{12}$ bits.

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage: $((n^2)^3)^2$ bits $= n^{12}$ bits.

For $n = 100$:

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage: $((n^2)^3)^2$ bits $= n^{12}$ bits.

For $n = 100$: 100^{12} bits

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage: $((n^2)^3)^2$ bits $= n^{12}$ bits.

For $n = 100$: 100^{12} bits $= 10^{24}$ bits

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage: $((n^2)^3)^2$ bits $= n^{12}$ bits.

For $n = 100$: 100^{12} bits $= 10^{24}$ bits $> 10^{21}$ bytes

What does this mean in practice?

Assume the best: Linear algebra on degree 3 suffices.

Practical time complexity **more** than:

$$((n^2)^3)^{2.5} = n^{15} \ .$$

n	n^{15}	$2^{n/2} \cdot n^2$
10	2^{50}	2^{12}
100	2^{99}	2^{64}
200	2^{114}	2^{116}

Storage: $((n^2)^3)^2$ bits $= n^{12}$ bits.

For $n = 100$: 100^{12} bits $= 10^{24}$ bits $> 10^{21}$ bytes $= 1000^7$ bytes $=$ 1 Zettabyte !