An easy attack on AEZ

Xavier Bonnetain  Patrick Derbez  Sébastien Duval  Jérémy Jean
Gaëtan Leurent  Brice Minaud  Valentin Suder

FSE 2017 Rump Session
Lightweight cryptography is required for the IoT

Here is a concrete example:

Toilet in my hotel is remote controlled!

Some models use Bluetooth!

Important confidentiality and authenticity issues!

Man in the 🚽 attack!

Denial of 🙃 attack!

Targeted attacks:

Welcome to the Internet of 🙃!
Lightweight cryptography is required for the IoT

Here is a concrete example:

- Toilet in my hotel is remote controlled!
- Some models use Bluetooth!
- Important confidentiality and authenticity issues!
- Man in the 🚽 attack!
- Denial of 🧼 attack!
- Targeted attacks:
- Welcome to the Internet of 🧼!
Cryptography for the Internet of Things

- Lightweight cryptography is required for the IoT

- Here is a concrete example:

- Toilet in my hotel is remote controlled!
- Some models use Bluetooth!

- Important confidentiality and authenticity issues!
- Man in the \(\text{\T{\front}}\) attack!
- Denial of \(\text{\T{\front}}\) attack!
- Targeted attacks: 

- Welcome to the Internet of \(\text{\T{\front}}\)!
Lightweight cryptography is required for the IoT

Here is a concrete example:

Toilet in my hotel is remote controlled!

Some models use Bluetooth!

Important confidentiality and authenticity issues!

Man in the 🚽 attack!

Denial of 👸 attack!

Targeted attacks:

Welcome to the Internet of 👸!
Cryptography for the Internet of Things

- Lightweight cryptography is required for the IoT

- Here is a concrete example:

- Toilet in my hotel is remote controlled!
- Some models use Bluetooth!

- Important confidentiality and authenticity issues!

- Man in the 🚿 attack!

- Denial of 🚿 attack!

- Targeted attacks:

- Welcome to the Internet of 🚿!
Cryptography for the Internet of Things

- Lightweight cryptography is required for the IoT

- Here is a concrete example:
  - Toilet in my hotel is **remote** controlled!
  - Some models use **Bluetooth**!

- Important **confidentiality** and **authenticity** issues!
- Man in the 🚿 attack!
- Denial of 👾 attack!

- Targeted attacks:

- Welcome to the Internet of 👾!
Cryptography for the Internet of Things

- Lightweight cryptography is required for the IoT

- Here is a concrete example:
  - Toilet in my hotel is remote controlled!
  - Some models use Bluetooth!

- Important confidentiality and authenticity issues!
  - Man in the 🧦 attack!
  - Denial of 👹 attack!

- Targeted attacks:

  Welcome to the Internet of 👹!
Cryptography for the Internet of Things

- Lightweight cryptography is required for the IoT
- Here is a concrete example:
  - Toilet in my hotel is remote controlled!
  - Some models use Bluetooth!
- Important confidentiality and authenticity issues!
- Man in the 🚿 attack!
- Denial of 👾 attack!
- Targeted attacks:
- Welcome to the Internet of 👾!
Lightweight cryptography is required for the IoT. Here is a concrete example:

Toilet in my hotel is remote controlled!

Some models use Bluetooth!

Important confidentiality and authenticity issues!

Man in the attack!

Denial of attack!

Targeted attacks:

Welcome to the Internet of ⚡!

Security company Trustwave issued an advisory to owners of the Satis smart toilet, currently available in Japan, which pairs with a Google Android app through Bluetooth.

Thanks to a vulnerability within the app, Trustwave says any person could download the My Satis app and gain control of any Satis smart toilet.

So, what potty perk could a Satis toilet owner discover? Trustwave says users could repeatedly flush the toilet, driving up water costs, or remotely open and close the toilet lid as well as activate the bidet, "causing discomfort or distress to (the) user."

The company says Satis makers Uji have yet to respond to the security issue.

According to a Satis brochure, the toilet features multiple cleansing options, heated seat and even plays music. NPR notes the Satis app allows users to control the commode and keep track of their bathroom activity.

This isn't the first case of a smart technology prone to harassment from curious Internet users. Last month, a Forbes writer detailed how she was able to gain easy access to home automation systems.
Viet Tung Hoang, Ted Krovetz & Phillip Rogaway
Robust Authenticated-Encryption
AEZ and the Problem That It Solves
EUROCRYPT 2015

▶ Very strong security goal: robust authenticated encryption
▶ Very complex design: huge state, many subcases

▶ Third round CAESAR candidate
▶ Tor is considering using AEZ
Figure 5: Illustration of AEZ enciphering. Rectangles with pairs of numbers are tweakable block ciphers, the pair being that tweak (the key, always K, is not shown). Top row: enciphering a message M of (32 or more bytes) with AEZ-core. The i-block (top left) is used for the bulk of the message, but the xy-block (top right) comprises the last 32 bytes, while the uv-block (top middle) comprises the prior 0–31 bytes. (The picture shows a uv-block of 17–31 bytes.) Bottom left: AEZ-hash computes \( \Delta = \oplus \Delta_i \) from a vector-valued tweak encoding \( A, N \), and a bytes. It’s i-th component \( T_1 \cdots T_m \) is hashed as shown. Bottom right: AEZ-tiny, when operating on a string M = L \( \| \) R of 16–31 bytes. More rounds are used if M has 1–15 bytes. In AEZ10. Then the design starts to seem like major overkill: each block \( M_i \) is subjected to 30 rounds of AES (ten shared with a neighboring block), plus additional AES rounds to produce the unpredictable, \( M \)-dependent value S that gets injected into the process while 20 rounds yet remain. In light of such overkill, AEZ-core selectively prunes some of the AES calls that AEZ10 would perform, using AES4 in their place. In particular, we prune invocations where we are trying to achieve computational xor-universal hashing. We leave enough AES rounds so that each block \( M_i \) is effectively processed with 12 AES rounds, eight of these subsequent to injection of the highly-unpredictable S and four of them shared with a neighboring block. The key steps in calculating S are not pruned, nor the TBC used to mask u-and v-blocks. Tweak processing. So far we have not mentioned the processing of the tweak T built from N.
Previous results on AEZ

- AEZv3: birthday attack recovers the key  
  [Asiacrypt 2015]
- Patched in AEZv4
  - Using Blake2 for key derivation
  - Bigger is better?
- AEZv4: birthday attack recovers the key  
  [FSE 2017]
Previous results on AEZ

- AEZv3: birthday attack recovers the key [Asiacrypt 2015]
- Patched in AEZv4
  - Using Blake2 for key derivation
  - Bigger is better?
- AEZv4: birthday attack recovers the key [FSE 2017]
Previous results on AEZ

- **AEZv3**: birthday attack recovers the key  
  [Asiacrypt 2015]

- Patched in AEZv4
  - Using Blake2 for key derivation
  - Bigger is better?

- **AEZv4**: birthday attack recovers the key  
  [FSE 2017]
AEZ-MAC (PMAC variant)

- With empty message, AEZ turns into a MAC

**AEZv3**

```
A1  △1  E  △3
   △2   E
   △3   E
```

**AEZv4**

```
A1  △1  E  △3
   △2   E
   △3   E
```
**XEX construction**

- $E(P \oplus \Delta_i) \oplus \Delta_i$ is a tweakable block cipher if $i \mapsto \Delta_i$ is an $\varepsilon$-AXU function.

- **Common constructions** ($L = E_k(0)$)
  - $\Delta_i = i \cdot L$ (OCB1, OCB3)
  - $\Delta_i = 2^i \cdot L$ (OCB2)

- **AEZv3** (subkeys $J$, $L$)
  - $\Delta_i = 8 \cdot J \oplus (i \mod 8) \cdot J \oplus 2^{\lfloor (i-1)/8 \rfloor} \cdot L$

- **AEZv4** (subkeys $J$, $L$)
  - $\Delta_i = L \oplus \left( 2^{3+\lfloor (i-1)/8 \rfloor} + (i - 1 \mod 8) \right) \cdot J$
A closer look

**AEZv4 offsets**

\[ \Delta_i = L \oplus \left( 2^{3+\lceil (i-1)/8 \rceil} + (i - 1 \mod 8) \right) \cdot J \]

- **Addition between GF(2^{128}) elements?**
  - \( \Delta_i = L \oplus 2^{3+\lceil (i-1)/8 \rceil} \cdot J \oplus (i - 1 \mod 8) \cdot J \)
    - \( 2^x \) is actually \( \alpha^x \), with \( \alpha \) a generator (\( \alpha^{128} = \alpha^7 \oplus \alpha^2 \oplus \alpha \oplus 1 \))
    - \( (i - 1 \mod 8) \) is one of \{0, 1, \( \alpha \), \( \alpha \oplus 1 \), \( \alpha^2 \), \( \alpha^2 \oplus 1 \), \( \alpha^2 \oplus \alpha \), \( \alpha^2 \oplus \alpha \oplus 1 \)\}

- **Is it injective?**
  - No!
  - \( \Delta_{40} = L \oplus \alpha^7 \cdot J \oplus (\alpha^2 \oplus \alpha \oplus 1) \cdot J \)
  - \( \Delta_{1001} = L \oplus \alpha^{128} \cdot J = L \oplus (\alpha^7 \oplus \alpha^2 \oplus \alpha \oplus 1) \cdot J \)
A closer look

AEZv4 offsets

$$\Delta_i = L \oplus \left(2^{3+[\frac{(i-1)}{8}]} + (i - 1 \mod 8)\right) \cdot J$$

- Addition between GF($2^{128}$) elements?
  - $$\Delta_i = L \oplus 2^{3+[\frac{(i-1)}{8}]} \cdot J \oplus (i - 1 \mod 8) \cdot J$$
    - $2^x$ is actually $\alpha^x$, with $\alpha$ a generator ($\alpha^{128} = \alpha^7 \oplus \alpha^2 \oplus \alpha \oplus 1$)
    - $(i - 1 \mod 8)$ is one of $\{0, 1, \alpha, \alpha \oplus 1, \alpha^2, \alpha^2 \oplus 1, \alpha^2 \oplus \alpha, \alpha^2 \oplus \alpha \oplus 1\}$

- Is it injective?
  - No!
  - $\Delta_{40} = L \oplus \alpha^7 \cdot J \oplus (\alpha^2 \oplus \alpha \oplus 1) \cdot J$
  - $\Delta_{1001} = L \oplus \alpha^{128} \cdot J = L \oplus (\alpha^7 \oplus \alpha^2 \oplus \alpha \oplus 1) \cdot J$
AEZv4 offsets

\[ \Delta_i = L \oplus \left( 2^{3 + \lfloor (i-1)/8 \rfloor} + (i - 1 \mod 8) \right) \cdot J \]

- Addition between \( GF(2^{128}) \) elements?
  - \( \Delta_i = L \oplus 2^{3 + \lfloor (i-1)/8 \rfloor} \cdot J \oplus (i - 1 \mod 8) \cdot J \)
    - \( 2^x \) is actually \( \alpha^x \), with \( \alpha \) a generator (\( \alpha^{128} = \alpha^7 \oplus \alpha^2 \oplus \alpha \oplus 1 \))
    - \( (i - 1 \mod 8) \) is one of \( \{0, 1, \alpha, \alpha \oplus 1, \alpha^2, \alpha^2 \oplus 1, \alpha^2 \oplus \alpha, \alpha^2 \oplus \alpha \oplus 1 \} \)

- Is it injective?
  - No!
  - \( \Delta_{40} = L \oplus \alpha^7 \cdot J \oplus (\alpha^2 \oplus \alpha \oplus 1) \cdot J \)
  - \( \Delta_{1001} = L \oplus \alpha^{128} \cdot J = L \oplus (\alpha^7 \oplus \alpha^2 \oplus \alpha \oplus 1) \cdot J \)
Conclusion

Forgery attack

- Swap $A_{40}$ and $A_{1001}$ $\leadsto$ same tag
- Swap $P_{79,80}$ and $P_{2001,2002}$ $\leadsto$ $C_{79,80}$ and $C_{2001,2002}$ swapped

- Similar to OTR attack
- Easy to patch: AEZv5?

- Even provably secure ciphers can be broken!

- Don’t use AEZv4 to secure your toilet!