This Week in Crypto: "Practical State Recovery Attack on ICEPOLE"*

by TAO HUANG, HONGJUN WU, IVAN TJOAWINATA

Premise: ICEPOLE is one of the submissions to the CAESAR. CAESAR is a competition to choose a new standard in Authenticated Encryption. It claims to be an online cipher, that has “immediate robustness” to nonce-misuse (since there is a secret number), is parallelizable and offers intermediate tags (so authentication can also be online).

ICEPOLE: For the complete specifications, one should refer to the CAESAR competition entry from Morawieki et. al., however, here’s enough of a summary of the cipher to understand the gist of the attack.

ICEPOLE has a sponge construction (like the one in SHA-3). Central to the construction is a 1280-bit state $S$. It is an array of bits of the form $S[4] [5] [64]$, or an array of 64-bit words $S[4] [5]$. The cipher takes associated data, $\sigma_{AD}^{0} ... \sigma_{AD}^{m}$, plaintext $\sigma_{P}^{0} ... \sigma_{P}^{n}$, a key, a nonce and optionally a secret message number $\sigma_{SMN}$ (dependant on which variant). It will output a ciphertext $c_{0} ... c_{n}$, and a authentication tag $T$.

For the permutation $P$, performed on $S$, it takes the form $P_{6}$ or $P_{12}$, which is 6 or 12 rounds of the round function $R$, respectively. $R$ has 5 constituents, taking the form $R = \kappa \circ \psi \circ \pi \circ \rho \circ \mu$. These, respectively, are:

- $\mu$: Multiplication by a constant matrix done on each column vector of 4 words.
- $\rho$: Bitwise rotation
- $\pi$: reordering of bits
- $\psi$: an SBox layer involving a 5-bit SBox.
- $\kappa$: XOR with constants

ICEPOLE's state is initialized with constants, then the nonce and key are mixed into the state by bitwise XOR with particular words in the state.

The tag is made up on two words extracted from the final state.

Differential-Linear Distinguishing Attack: This attack assumes that the nonce and $\sigma_{SMN}$ can be reused as the attacker chooses. It is a combination of a differential attack and a linear attack, meaning that we use messages of certain input difference and analyze the statistics of the differences of chosen bits (according to a linear mask) in the output. This makes use of $P_{0}$.

To find useful differential characteristic, the authors note that the SBox is the only non-linear operation, therefore, considering the input and output of the SBoxes is sufficient. This is equivalent to studying the number of active SBoxes. Using a SAT-solver, there is a minimum of

---

*This Week in Security

@ruthng

ruthng@thisemailisnottruthless@gmail.com

Page 1
9 active SBoxes over 3 rounds. However, this differential path is not feasible in the cipher itself. The authors then show, from this, that the minimum number of active SBoxes in the first round is 2, leading to the generation of 5 possible initial differences with good statistical probabilities over 3 rounds, which they name $D_1, \ldots, D_5$.

To find useful linear characteristics, the authors note that the SBox itself has maximum bias of $\frac{1}{4}$. They also required that the weight of the output mask be minimized, leading to two good linear characteristics over 3 rounds, denoted $L_1, L_2$.

The authors then needed to concatenate the differential and linear characteristics into a 6-round differential-linear characteristic. They made use of $D_2$, the only one with two active SBoxes in the last row, that has known bits. Then, they considered all possible rotations of $L_1, L_2$ with respect to random pairs of PT blocks with the chosen initial difference in $D_2$. They found that $L_1$ rotated by 33 bits had the best results.

From this, the authors complete the description of their distinguishing attack. Making use of the characteristic above, they require $2^{31.6}$ PT-CT pairs with the difference $D_2$. After encrypting the block, in $2^{31.6}$ of the cases, the bits at position $S[1][1][33]$ in round 5 can be recovered from the bits in the CT (round 6). For the remaining $2^{31}$ PT-CT pairs where both bits in the pair were recovered, if the XORed differences of the pairs of bits at that position show a bias of more than $2^{-10.2}$, we can say with 99.3% certainty that we have distinguished ICEPOLE.

**State Recovery Attack:** The authors then present a state recovery attack, which will allow an adversary to decrypt messages and forge tags. This suggests that if the nonce and key are kept constant across sufficient rounds, ICEPOLE is not secure.

The attack begins by noting that there are only 256 unknown bits in the state before the first $P_6$. These are 4 64-bit unknown words in the last column. Denote this $U_0, U_1, U_2, U_3$.

The attack begins by recovering $U_0, U_3$, which starts by recovering $S[1][1][33]$ for pairs of PT-CT with the differential $D_2$. Retain the cases where $S[1][1][33]$ are consistent across the pair, then make use of the inherent bias in the SBox, and this subset of pairs (which we know exhibit a bias), to retrieve two linear equations relating bits $U_0^{31}, U_1^{31}, U_2^{31}, U_3^{31}$, derived from tracing back the active SBoxes.

The attack then rotates the differential-linear characteristic to derive similar equations for the 63 other possible rotations, thereby retrieving $U_0, U_3$. Similarly, $U_2$ can be retrieved by starting with $D_1$ and $L_2$, and solving for $S[3][1][58]$, and $U_4$ can be retrieved by starting with $D_4$ and $L_2$, solving for $S[3][1][35]$. These also can be rotated to find the entire of $U_2, U_4$.

From here, the entire state has been retrieved with probability around 99%. To be more confident about this, the authors suggest verifying with additional PT-CT pairs. They also note that one can allow for up to 7 errors by exhaustively bit-flipping up to 7 bits to find the correct key, to increase the probability of success to 99.9995%. This comes at negligible additional complexity. In total, the main attack requires $2^{34.8}$ encryptions and, for the verification step, another $2^{37.5}$. It has $2^{45.8}$ data complexity.

**Implications:** Note that while the attacker here retrieves the state (and not the secret number/key), this is sufficient to forge tags and decrypt future traffic that makes use of this secret number-key pair. The authors programmed this attack as a proof of concept, and were able to correctly retrieve the state, with only one bit of error. Finding $U_0, U_3$ took 15.3 hours, finding $U_2$ took 3.5 days and $U_1$ took 3.5 days. The first of these was done on a server with 48 cores, while the latter two were on a server with 64 cores.

---

**This Week in Security: “The Matter of Heartbleed”**

_by Zakir Durumeric, James Kasten, David Adrian, J. Alex Halderman, Michael Bailey, Frank Li, Nicholas Weaver, Johanna Amann, Jethro Beekman, Mathias Payer, Vern Paxson_

**Premise:** Heartbleed is a vulnerability discovered in 2014, which, in addition to putting large amount of private data at risk, also revealed how the worldwide community reacts and manages such a breach in security. The authors here evaluated the timeliness and effectiveness of the global community’s response to this bug.

**Heartbleed bug:** The Heartbleed vulnerability was discovered in the TLS Heartbeat Extension to OpenSSL. In short, it can be requested by peers during the initial TLS handshake, and after the session has been initialized, each endpoint can send a `HeartbeatRequest()` to verify connectivity. The vulnerability is that this allowed either end point to read its peer’s memory by specifying a payload length larger than the amount of data in the `HeartbeatRequest()` message. This is because the peer trusts the attacker-specified length of an attacker-controlled message. Thus, the peer can transmit up to $2^{16}$ bytes of data to the attacker unknowingly, the maximum length that the attacker can specify in the `HeartbeatRequest()`. This could include passwords, private keys to sign certificates or other sensitive personal information.

**Vulnerability Scans:** The authors performed vulnerability scans against the Alexa Top 1 Million domains and against 1% of the public, non-reserved, IPv4 address space, at regular intervals. In their scans, they sent `HeartbeatRequest()` to the websites they were interested in with a length field set to zero (restricted, of course, to websites that still supported the Heartbeat Extension). If the Heartbeat Extension has been patched, these requests would be rejected, however, the vulnerable version would respond with a message filled only with...
useless padding (since 0 bytes were requested). This allowed the researchers to conduct their research without breaching any privacy boundaries of the websites they were looking at.

The first thing the authors realized was that all of Alexa’s Top 100 websites were patched within 48 hours of disclosure. Some of these websites had been informed prior to the official vulnerability disclosure. They do also reference another scan done 22 hours after the disclosure of the top 10000 domains, of which 630 sites remained vulnerable, including Yahoo, Imgur and some other rather popular sites.

The authors also scanned the top 1 million websites, revealing that 48 hours after disclosure, 45% supported HTTPS, of which 60% supported the Heartbeat Extension, and 11% were vulnerable to Heartbleed. They also showed that 91% of the HTTPS sites were powered by web servers that were known to be vulnerable at the time, which did not know about the bug prior to the disclosure. This creates an upper bound on the percentage of the HTTPS-enabled sites in the top 1 million sites that were vulnerable to the Heartbleed bug at 55%, when the vulnerability was disclosed. They also note that if certain sites were using versions of TLS from before the extension was patched in, they would not be vulnerable to Heartbleed. By studying SSL-Pulse statistics, overall statistics published about websites that use SSL, they showed that 32.6% of SSL sites run TLS 1.1 or 1.2, which were before the Heartbeat Extension. This gives a lower bound of 23.7% of the HTTPS-enabled sites in the top 1 million sites that were vulnerable to Heartbleed. Therefore, we can estimate that at the time of disclosure between 23.7% and 55% of the HTTPS-enable sites within the top 1 million sites were vulnerable to Heartbleed.

The authors also scanned 1% of the public IPv4 address space, beginning 48 hours after disclosure of the bug, daily. The first scan showed that 11.4% of HTTPS hosts supported the Heartbeat Extension and 5.9% of all HTTPS hosts were vulnerable. They also noted that 10 Autonomous Systems (ASes) accounted for 50% of the vulnerable hosts but represented only 8.6% of all HTTPS hosts.

The authors also studied embedded systems that may be affected by Heartbleed, such as VPNs, printers security cameras and the like. They analyzed the self-signed certificates employed by the hosts to group the systems found into large clusters then manually inspected the clusters. The authors were conservative in their study, thereby minimizing false positives. However, they did derive 74 sets of vulnerable devices. These range from Cisco video conferencing products to Dell printers.

The authors finally studied the impact of Heartbleed on services other than public HTTPS web services. This takes the form of mail servers, the Tor network, Bitcoin network, Android and wireless networks. Interesting results of this include that 7.5% of web servers (lower bound) supporting SMTP and TLS were vulnerable to Heartbleed, 48% of Tor networks were vulnerable, and that Bitcoin needed to suspend its services to patch the system, after US$6500 worth of bitcoins were stolen.

Certificate Ecosystem: One of the things that Heartbleed might have exposed are private keys with which certificates were signed. Therefore, the authors also looked at how speedily certificates got replaced or revoked.

Of the top 1 million sites that were vulnerable in the scan 48 hours after disclosure, only 10.1% replaced their certificates in the next month, even though 73% patched their SSL during the same time frame, suggesting most people patched the SSL Extension without replacing certificates. In addition, only 19% of the sites that replaced their certificate also revoked the original certificate, and 14% re-used the same private key to replace the certificate, essentially defeating the point of replacing their certificate at all.

However, Certification Authorities (CAs) did note that there were as many revocations in the three months following Heartbleed as there had been in the past 3 years. Individual CAs revoked large numbers of certificates (such as GlobalSign revoking 50.2% of their visible certificates). This revoke caused large burdens on clients as well as servers since the large Certificate Revocation Lists had to be downloading to clients, CloudFlare, for example, had a list which had expanded from 2KB to 4.7MB. This required an additional 40Gbps of traffic per month, costing US$400,000 per month.

The authors also noted that changing or revoking a certificate did not stop the exploitation of previously recorded traffic. This suggested the need for forward secrecy. However, the authors found that 44% of the connections observed used cipher suites offering forward secrecy and that this has not risen appreciably in the wake of Heartbleed at all.

Attack Scene: The authors also took an interest in increased attempts to exploit the Heartbleed vulnerability after it was exposed. This was done via a honeypot at Amazon EC2, and existing passive taps of traffic. In particular, they studied the pattern of setting up TLS sessions to find attackers who exploited the system prior to the session was done setting up, and studied the length of encrypted messages after the session was set up to look for Heartbeat messages that are malicious.

Apart from two Heartbleed scanning services, filippio.io and sslabs.com, four sources of attacks were found that targeted more than 100 addresses in the ICSI network (a network set up at UC Berkeley). Two of these were located in CHINANET, one at Nagravision and one at Rackspace. These attacks targeted HTTPS ports, but there were also small numbers of exploits on mail servers and other services.

The authors found that majority of attacks did not target many sites, suggesting targeted attacks and not wide-spread scanning. However, a whopping 201 sources attempted to exploit the honeypot at Amazon EC2, sug-
gesting that adversaries preferentially scanned denser address spaces, such as those at Amazon EC2.

**Notifications:** Finally, the authors extracted email contacts from the WHOIS record of the vulnerable hosts that were scanned and contacted them regarding the study and the vulnerability discovered. This excluded embedded systems, and those that were managed by larger organizations (e.g. Amazon).

Hearteningly, 57% of the people contacted patched their systems within a couple of weeks. Statistical analysis showed that there was no language barrier since websites that responded in English had a roughly equal patch rate to those that did not. Those that had human responses, or automated responses saw the highest patch rate, with 47% and 32% patching within one day respectively. Overall, 77% of contacts never responded, but 20% of these also patched within one day.

**Discussion:** This has incited debate on a few levels:

- The fact that many administrators did not replace certificates after patching, or reused their private key, suggests that many administrators did not have a good understanding of the bug or of the HTTPS system as a whole, which is worrying for the response to future threats.

- The widespread effect of the bug was precipitated by the fact that many applications that had no use for the extension continued to enable the extension in their web services. This suggests the need to relook at how extensions are managed and used. In addition, this suggests that insufficient code review was done on an extension that was to be widely adopted.

- The eruption in traffic regarding revocation put unprecedented strain on CAs, suggesting the need for a model of CAs that allowed for more scalability, since this may be unsustainable as the web continues to grow.

- The mess of advance notifications led to a multitude of important websites not being patched ahead of the public announcement, leaving them vulnerable to opportunistic and quick-acting adversaries. Therefore, more work should be done to balance the effects of advance disclosure with that of premature leaks so that the disclosure does not lead to substantial breaches in privacy.

- The authors’ work on notifying individuals about the vulnerability suggests that there is a high response rate to direct notifications. Thus, future attempts to disseminate reports on vulnerabilities can be more targeted, so that more services can be patched more quickly.