Related-Key Forgeries for Prøst-OTR

Christoph Dobraunig, Maria Eichlseder, and Florian Mendel

IAIK, Graz University of Technology, Austria
maria.eichlseder@iaik.tugraz.at

Abstract. We present a forgery attack on Prøst-OTR in a related-key setting. Prøst is a family of authenticated encryption algorithms proposed as candidates in the currently ongoing CAESAR competition, and Prøst-OTR is one of the three variants of the Prøst design. The attack exploits how the Prøst permutation is used in an Even-Mansour construction in the Feistel-based OTR mode of operation. Given the ciphertext and tag for any two messages under two related keys K and $K \oplus \Delta$ with related nonces, we can forge the ciphertext and tag for a modified message under K. If we can query ciphertexts for chosen messages under $K \oplus \Delta$, we can achieve almost universal forgery for K. The computational complexity is negligible.

Keywords: CAESAR competition, cryptanalysis, Prøst, authenticated encryption, related-key

1 Introduction

Due to the currently ongoing CAESAR competition for authenticated encryption [23], the new favourite toy of the cryptographic community are clearly authenticated ciphers. A significant collective effort will be necessary to judge the 57 submitted candidate ciphers with respect to their security and applicability. The goal of this cryptographic competition is to identify a portfolio of reliable, efficient, secure authenticated encryption algorithms with unique features for different application scenarios. Experience with previous competitions and focused projects like AES, SHA-3, eSTREAM and NESSIE has clearly demonstrated that the joint effort of the community to focus on a particular topic can impressively advance the understanding of the reasearched primitives in a relatively short period of time. Right now, first security analyses of the submitted candidates are necessary to allow the competition committee to judge the first-round candidates adequately, and select the most promising submissions for the next round.

Prøst, designed by Kavun et al. [15], is one of the candidates submitted to the CAESAR competition. It combines a newly designed, efficient permutation, the Prøst permutation, with several modes of operation. The resulting Prøst family of authenticated ciphers consists of three variants: Prøst-COPA, Prøst-OTR, and Prøst-APE, each with its own advantages and features. The Prøst-OTR variant uses the Prøst permutation in a single-key Even-Mansour construction [9,11,12]

as a block cipher in Minematsu's provably secure, Feistel-based OTR mode of operation [20]. Due to the novelty of the design, previous cryptanalysis results on Prøst itself are limited to the designers' own analysis, published together with the design document [15].

We present a forgery attack on Prøst-OTR in a related-key setting. The scenario is that an attacker is given ciphertexts and tags of two messages: one under the target key K, and one under a related key $K \oplus \Delta$ for some arbitrary Δ. Both keys are secret, but their difference Δ is known to the attacker. The nonces used for encrypting the two messages are also related in a similar way. Then, with negligible computational complexity, the attacker can forge the ciphertext and authentication tag for a third message under the target key K. In fact, depending on the length of the original messages, forgeries for a large number of fake messages can be obtained. In addition, in case the attacker has control over one of the two originally encrypted messages, he can even control the content of the third, forged message.

Our attack is generic and exploits the combination of the OTR mode of operation with an Even-Mansour block cipher construction. It is independent of the used permutation, and thus does not use any particular properties or weaknesses of the Prøst permutation. Consequently, the other members of the Prøst family, Prøst-COPA and Prøst-APE, are not affected or endangered by the attack. However, the attack demonstrates the possible complications of using an Even-Mansour construction as a block cipher in otherwise secure modes of operation. The Even-Mansour approach of creating a block cipher from a pseudorandom permutation by xoring a secret key before and after applying the permutation to the plaintext has been studied extensively [6,7,8,9,13,18]. It has been proven secure under different notions of security, with detailed bounds relating the security level with the key length. However, it is inherently susceptible to related-key attacks. The OTR mode of operation allows to "lift" this property to the full encryption and authentication scheme. This unfortunate combination of otherwise secure building blocks shows two things: that the Even-Mansour construction should only be used very cautiously, and that related-key properties are not well covered by the classical security notions, although they can lead to powerful forgery attacks.

Related-key setups are a relatively strong attack setting. Nevertheless, depending on the exact requirements, they are often not entirely far-fetched in practical scenarios. In particular, scenarios where only a known (but arbitrary) difference Δ between any two unknown keys is required, like in our attack, are quite realistic, and occur as side effects of several published protocols. The only limitation the attack imposes on Δ is that it does not affect the least significant bits of the key. For compatibility with the nonce difference, the modified part of the key must not be longer than the nonce length (half the key size in Prøst-OTR).

As an example for related keys in practice, consider the WEP standard [14]. There, the keys for the individual communication links are derived by concatenating (public, random) IVs with the fixed secret WEP key. Clearly, any two keys

constructed this way have a publicly known differential relation. Similar scenarios could be imagined in any other network of resource-constrained devices (e.g., of sensor nodes), where individual encryption keys need to be derived in a cheap way from some master secret (e.g., by xoring individual IDs, nonces or challenge values to the key). Despite its inherent susceptibility to birthday attacks, the idea to "xor nonce to key" is also incorporated in several CAESAR candidates, such as AVALANCHE [1] and Calico [22]. Recently, cheap modifications of some master secret have also gained some popularity as a countermeasure to side-channel attacks, termed "fresh re-keying". The rationale is that to avoid differential side-channel attacks, subsequent encryption processes should never use the same key twice, but derive some sort of session keys from the long-term key in a cheap way.

The additional requirement of related nonces is not as strong as the related keys. In many applications, nonces are generated in a very predictable pattern (typically a simple counter as a message sequence number). In some cases, the attacker may even be able to influence the nonce counter: a simple example is by triggering encryptions until the nonce counter arrives at the desired value, or by somehow causing the device to jump the unwanted nonce values. We note that the attack does not require "nonce misuse" in the sense that the attacker requests repeated encryptions under the same nonce.

Related-key attacks [4,16] have been studied extensively, for various ciphers and applications. A prominent example is Biryukov et al.'s related-key attack on AES [5], which makes very strong assumptions about the relations between subkeys. The combination of related keys with related nonces has previously been applied primarily to stream ciphers, in particular in the context of the eSTREAM project. Examples include the key recovery attacks on Grain-v1 and Grain-128 by Lee et al. [19], or the recent analysis of generic chosen-IV attacks with applications to Trivium by Pasalic and Wei [21].

Outline. We first describe the Prøst family of authenticated ciphers and the notational conventions for the remaining document in Section 2. In Section 3, we derive a first basic related-key attack on Prøst-OTR. In Section 4, we propose a few possible improvements to the attack and extended attack scenarios. Finally, in Section 5, we conclude with a discussion of the applicability of the Prøst-OTR attack to other authenticated encryption modes.

2 Description of Prøst-OTR-n

2.1 The Prøst family of authenticated ciphers

Prøst is a family of authenticated encryption algorithms. Kavun et al. [15] proposed the cipher family as a candidate in the currently ongoing CAESAR competition [23] for authenticated ciphers. Prøst comes in three flavors: Prøst-COPA, Prøst-OTR and Prøst-APE. All flavors share the same core permutation, the

Prøst permutation designed by Kavun et al. [15], but use it in different modes of operation.

Prøst-APE uses the Prøst permutation in Andreeva et al.'s sponge-based APE mode [2]. The other two flavors, Prøst-OTR and Prøst-COPA, use modes of operation that are originally not permutation-based, but block-cipher-based: Andreeva et al.'s COPA mode [3], and Minematsu's OTR mode [20]. In these variants, the Prøst permutation is used in a single-key Even-Mansour construction [9] to provide the required block cipher.

Each of the three flavors is available in two security levels, specified by a parameter $n \in \{128, 256\}$, resulting in a total of six proposed cipher family members. The designers rank the COPA variants as their primary recommendations, the OTR variants second, and the APE variants last.

2.2 Notation

Throughout this paper, we use essentially the same notation as Prøst's designers [15]. Unless noted otherwise, all operations are performed in $\mathbb{F}_{2^{2n}}$ with respect to Prøst's irreducible polynomial, where $n \in \{128, 256\}$ defines the security level. For convenience of notation, elements in $\mathbb{F}_{2^{2n}}$ are often represented interchangeably as elements of $\mathbb{F}_2^{2n}$. We denote addition in $\mathbb{F}_{2^{2n}}$ (xor) by $\oplus$, and multiplication in $\mathbb{F}_{2^{2n}}$ by $\cdot$ (operator omitted where possible). By $N\|10^*$, we mean the n-bit bitstring $N \in \mathbb{F}_2^n$, concatenated with $(1, 0, \ldots, 0) \in \mathbb{F}_2^n$ to get an element in $\mathbb{F}_2^{2n}$. Otherwise, numbers mean integer numbers $\in \mathbb{Z}$ or individual bits $\in \mathbb{F}_2$ when written in roman font $(1, 2, 3, \ldots)$, but elements of $\mathbb{F}_2^{2n}$ in truncated hex notation when written in typewriter font $(\mathtt{1}, \mathtt{2}, \mathtt{3}, \ldots)$; for example, $\mathtt{13} = (0, \ldots, 0, 1, 0, 0, 1, 1) \in \mathbb{F}_2^{2n}$. The variable names we use are summarized in Table 1.

Table 1: Notation and variables used throughout this document.

n	security level
K, K'	$2n$-bit keys (related keys)
N	n-bit nonce
$M = M_0 \cdots M_{2m-1}$	the padded message, split into $2n$-bit blocks
$C = C_0 \cdots C_{2m-1}$	the ciphertext in $2n$-bit blocks
T	n-bit tag
ℓ	secret counter basis, derived from K and N ($= \delta$ in [15])
P	the Prøst permutation
$\tilde{P}_K$	P used in single-key Even-Mansour mode as block cipher
Σ	sum of message blocks, basis for the tag T
Δ	difference between the related keys K and $K' = K \oplus \Delta$
M', C', T'	message encrypted under related key K' and nonce
$\widetilde{M}, \widetilde{C}$	modified message and ciphertext
M^*, C^*, T^*	attacker's forged message, ciphertext and tag
α, γ	intermediate values, inputs to P

2.3 Prøst-OTR-n

Prøst-OTR-n uses the block cipher $\tilde{P}_K$, built from the permutation P in a single-key Even-Mansour construction [9], in Minematsu's OTR mode of operation [20]. The result is a nonce-based authenticated encryption scheme with online encryption and decryption that is fully parallelizable [15]. Prøst-OTR-n is proposed in two security levels, $n \in \{128, 256\}$. The security level defines the permutation size $2n$ and block size $2n$, the key size $2n$ and nonce size n, and the tag size n. The claimed security for Prøst-OTR-n is $\frac{n}{2}$ bits (confidentiality and integrity of plaintext and integrity of associated data). No particular claims are made for or against the related-key security of the cipher.

Since our attack does not exploit any particular properties of the Prøst permutation $P : \mathbb{F}_2^{2n} \to \mathbb{F}_2^{2n}$, we do not include the definition of P in this description. The design of the permutation-based block cipher $\tilde{P}_K$, however, is essential for the attack. For a key $K \in \mathbb{F}_2^{2n}$, the block cipher $\tilde{P}_K : \mathbb{F}_2^{2n} \to \mathbb{F}_2^{2n}$ is defined as follows:

$$\tilde{P}_K(x) = K \oplus P(x \oplus K).$$

In OTR, message blocks M_j are encrypted in pairs in 2-round Feistel networks to get the ciphertext blocks C_j. The Feistel round function first adds a counter-like value, then applies the block cipher $\tilde{P}_k$. For the counter-like value, a helper value ℓ is computed in an initialization phase by encrypting the padded nonce $N\|10^*$ under $\tilde{P}_K$. After processing all block pairs, the tag T is finally computed by encrypting a function of the checksum Σ, which is the xor of all odd-indexed message blocks M_{2i+1}. The detailed algorithm is listed in Algorithm 1 and illustrated in Fig. 1. For simplicity, we only describe the mode for empty associated data, and only for padded messages with an even (rather than odd) number of message blocks.

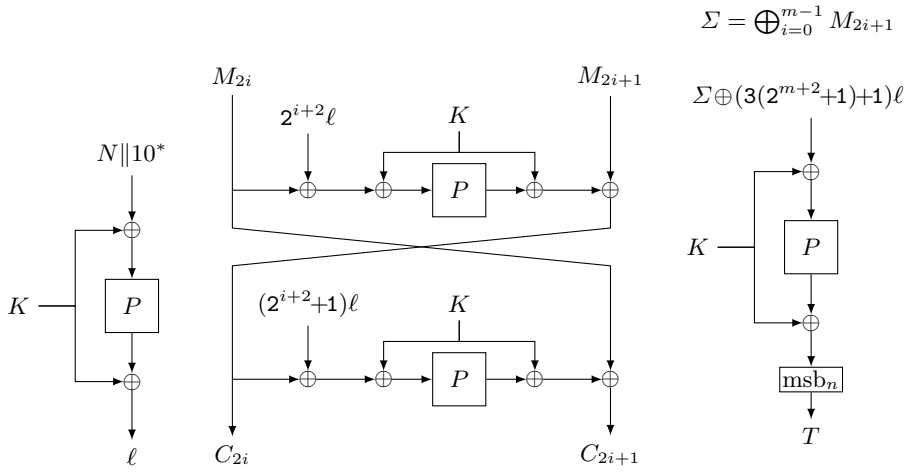
Algorithm 1 Prøst-OTR-n encryption

Input: padded message $M\|01^* = M_0 \cdots M_{2m+1}$, padded nonce $N\|10^*$
Output: ciphertext $C = C_0 \cdots C_{2m+1}$, tag T

$\Sigma \leftarrow \mathsf{0}$
$\ell \leftarrow \tilde{P}_K(N\|10^*)$
for $i = 0, \ldots, m-1$ **do**
  $C_{2i} \leftarrow \tilde{P}_K(\mathsf{2}^{i+2}\ell \oplus M_{2i}) \oplus M_{2i+1}$
  $C_{2i+1} \leftarrow \tilde{P}_K(\mathsf{2}^{i+2}\ell \oplus \ell \oplus C_{2i}) \oplus M_{2i}$
  $\Sigma \leftarrow \Sigma \oplus M_{2i+1}$
$T \leftarrow \mathrm{msb}_n(\tilde{P}_K(\mathsf{3}(\mathsf{2}^{m+2}\ell \oplus \ell) \oplus \ell \oplus \Sigma))$

3 Basic Forgery Attack on Prøst-OTR

In this section, we describe our basic forgery attack on Prøst-OTR. The attack exploits the combination of the OTR mode with the Even-Mansour block cipher



(a) Initialization (b) Message blocks M_{2i}, M_{2i+1}, $0 \leq i < m$ (c) Finalization

Fig. 1: Encrypting $2m$ message blocks M_j with Prøst-OTR-n under key K and nonce N. All values are $2n$ bits, with $n \in \{128, 256\}$, except the n-bit tag T.

construction, and is independent of the concrete permutation P used. We consider a related-key scenario, where encrypted messages of two different keys K and K' can be observed. Both K and K' are secret, but we assume the attacker knows the difference $\Delta = K \oplus K'$ (i.e., $K' = K \oplus \Delta$). In addition, we assume that the attacker can observe encrypted messages for related nonces N, N', such that $\Delta = (N\|10^*) \oplus (N'\|10^*)$. Since the last n bits of the padded nonces are identical, this means that the n least significant bits of Δ must be 0.

The basic idea of the proposed forgery attack is to combine information from the encryption of the same message M under the two related keys K, K' to forge a ciphertext and tag for a modified message M^* under one of the two keys, K. More specifically, we will first show how to use the ciphertext from the related key $K' = K \oplus \Delta$ to forge ciphertexts for modified messages under the target key K. Then, we will combine original and forged ciphertexts in a way such that the original tag remains valid for the resulting modified plaintext under K. The attack works for any plaintext of sufficient length ($\geq$ 514 message blocks for Prøst-OTR-128, $\geq$ 1026 blocks for Prøst-OTR-256).

3.1 Forging the ciphertext

Assume that the attacker obtains the ciphertext for the same message $M = M_0 \cdots M_{2m-1}$ (from Fig. 1) under a related key $K' = K \oplus \Delta$ and a related nonce $N'\|10^* = (N\|10^*) \oplus \Delta$, as illustrated in Fig. 2. Note that since the nonce only has length n (instead of $2n$ like the other values), Δ must only modify the most significant n bits, i.e., $\Delta = \Delta_n\|0^n$. Then, in the initialization phase illustrated in Fig. 2a, the differences in K' and N' cancel out right before the

call to the permutation P in the initialization. Thus, we receive a related counter value ℓ' with a simple relation to the original ℓ:

$$\begin{aligned}\ell' &= P_{K'}(N'\|10^*) = K' \oplus P((N'\|10^*) \oplus K') \\ &= K \oplus \Delta \oplus P(K \oplus \Delta \oplus (N\|10^*) \oplus \Delta) \\ &= \ell \oplus \Delta.\end{aligned}$$

Now consider the encryption of a modified message with message blocks

$$\widetilde{M}_j = M_j \oplus (2^{\lfloor j/2 \rfloor + 2} + 1)\Delta$$

under the original key K and nonce N. As Fig. 3 illustrates, the message differences "cancel out" with the corresponding difference in the ℓ values from the encryption under the related key in Fig. 2. Thus, in both Fig. 2 and Fig. 3, the inputs α and γ to the permutations are the same:

$$\begin{aligned}\alpha &= \widetilde{M}_{2i} \oplus 2^{i+2}\ell \oplus K \\ &= M_{2i} \oplus 2^{i+2}\ell \oplus 2^{i+2}\Delta \oplus \Delta \oplus K, \\ \gamma &= \widetilde{M}_{2i+1} \oplus P(\alpha) \oplus (2^{i+2} + 1)\ell \\ &= M_{2i+1} \oplus P(\alpha) \oplus 2^{i+2}\ell \oplus 2^{i+2}\Delta \oplus \ell \oplus \Delta.\end{aligned}$$

For this reason, the ciphertext $\widetilde{C}_j$ of the modified message block $\widetilde{M}_j$ under the original key K can be derived from the ciphertexts C'_j of the original message M_j under the related key $K \oplus \Delta$:

$$\begin{aligned}\widetilde{C}_{2i} &= \widetilde{M}_{2i+1} \oplus P(\alpha) \oplus K \\ &= C'_{2i} \oplus 2^{i+2}\Delta, \\ \widetilde{C}_{2i+1} &= \widetilde{M}_{2i} \oplus P(\gamma) \oplus K \\ &= C'_{2i+1} \oplus 2^{i+2}\Delta,\end{aligned}$$

since

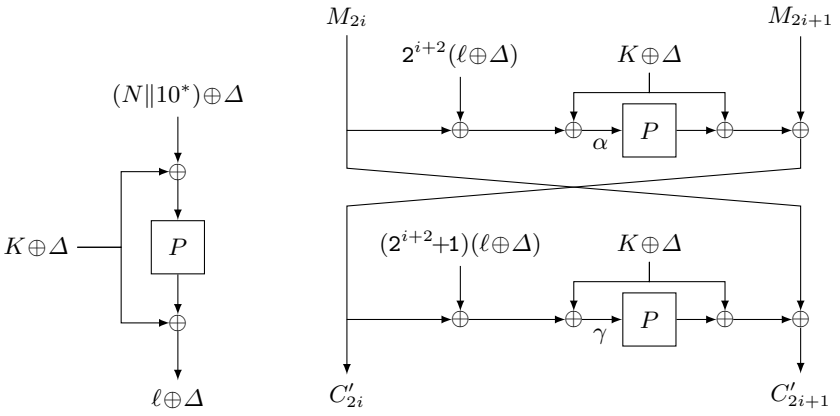
$$\begin{aligned}C'_{2i} &= M_{2i+1} \oplus P(\alpha) \oplus K \oplus \Delta, \\ C'_{2i+1} &= M_{2i} \oplus P(\gamma) \oplus K \oplus \Delta.\end{aligned}$$

Now, we know the correct ciphertexts for a modified message. However, we still need to find the corresponding authentication tag. We will try to re-use the original tag T for our forged message.

3.2 Forging the tag

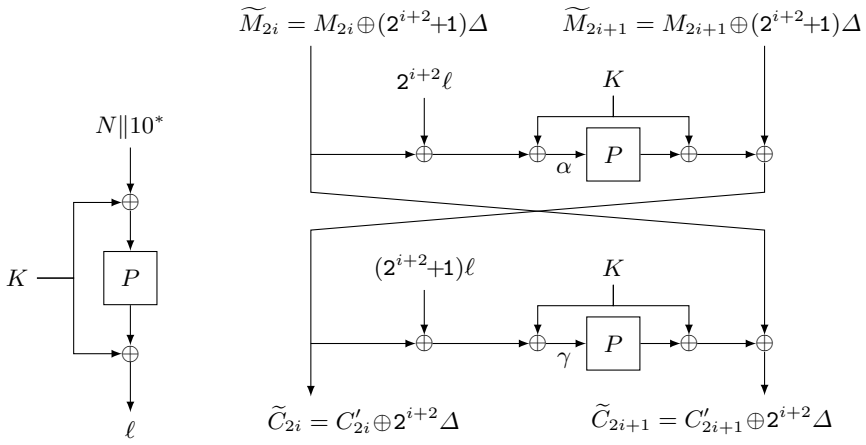
For a fixed key K and nonce N, the authentication tag only depends on the xor sum of all message blocks with odd index,

$$\Sigma = \bigoplus_{i=0}^{m-1} M_{2i+1}.$$



(a) Initialization (b) Message blocks M_{2i}, M_{2i+1}, $0 \le i < m$

Fig. 2: Encrypting the original message blocks M_j under a related key $K \oplus \Delta$ and nonce.



(a) Initialization (b) Message blocks $\widetilde{M}_{2i}, \widetilde{M}_{2i+1}$, $0 \le i < m$

Fig. 3: Encrypting modified message blocks $\widetilde{M}_j = M_j \oplus (2^{\lfloor j/2 \rfloor+2} + 1)\Delta$ under the original key K and nonce N.

Thus, if we want to re-use the original tag T for our forged message, we need to make sure that any induced differences cancel out when summing up the message blocks. We want to use original and modified message M and $\widetilde{M}$ to construct the final forged message M^* that satisfies this property.

For each message block pair M^*_{2i}, M^*_{2i+1} of the forged message M^*, we can decide to use either the original message block pair M_{2i}, M_{2i+1}, or the modified blocks $\widetilde{M}_{2i}, \widetilde{M}_{2i+1}$. Let λ_i denote whether we use the original ($\lambda_i = 0$) or modified ($\lambda_i = 1$) block pair for $0 \leq i < m$. Then, we get the message sum

$$\Sigma^* = \bigoplus_{i=0}^{m-1} M^*_{2i+1} = \Sigma \oplus \bigoplus_{i=0}^{m-1} \lambda_i(2^{i+2} + 1)\Delta.$$

Note that if Σ would sum up all message blocks (not only every second), then any choice of λ_i would create a successful forgery, since $M_{2i} \oplus M_{2i+1} = \widetilde{M}_{2i} \oplus \widetilde{M}_{2i+1}$. As it is, however, we need to select suitable coefficients $\lambda_i \in \mathbb{F}_2$ such that at least one coefficient λ_{i^*} is nonzero and

$$\bigoplus_{i=0}^{m-1} \lambda_i(2^{i+2} + 1)\Delta = 0. \tag{1}$$

Since $\{(2^{i+2} + 1)\Delta\} \subseteq \mathbb{F}_2^{2n}$, a vector space with dimension $2n$, any $2n + 1$ such vectors are linearly dependent, and suitable coefficients λ_i exist. Thus, for any given key difference Δ and known plaintext M with $2m \geq 4n + 2$ message blocks, we can solve this system of equations to find suitable coefficients λ_i. The ciphertext blocks C^* for the resulting forged message M^* can be computed as in Section 3.1, while the correct tag $T^* = T$ can be copied from M.

Summarizing, from observing the ciphertext and tag for encryptions of the same message M under two related keys K and $K' = K \oplus \Delta$, the attacker has forged the ciphertext C^* and tag T^* for a different message M^* of the same block length with negligible computational effort. The attacker knows this forged message, but has almost no control over its contents. The attack nonce is the same as the original nonce N. We discuss some remarks and improvements to this attack in Section 4.

3.3 Practical example

For illustration, we apply the attack to Prøst-OTR-128 with $n = 128$. This variant of Prøst-OTR uses a 256-bit key, a 128-bit nonce, and a message blocksize of 256 bits. The irreducible polynomial for the finite field $\mathbb{F}_{2^{2n}}$ is $f(x) = x^{256} \oplus x^{10} \oplus x^5 \oplus x^2 \oplus 1$.

As a simple example, assume that $\Delta = 2^{128}$. Then, the related key and nonce for the target key K and nonce N are

$$K' = K \oplus 2^{128},$$
$$N' = N \oplus 1.$$

Assume that some message M with 514 blocks of 256 bits each was encrypted under K to ciphertext C and tag T, and under K' to C' and T'.

For each block pair (M^*_{2i}, M^*_{2i+1}) of the forged message M^*, we now need to decide whether we copy the original message (M_{2i}, M_{2i+1}) or the modified version $(\widetilde{M}_{2i}, \widetilde{M}_{2i+1})$. Our choice needs to satisfy the coefficient equation (1). A solution can easily be found by hand; an example is given in Table 2.

Table 2: A solution for coefficients $\lambda_i = 1$ in equation (1) in $\mathbb{F}_{2^{256}}$ with field polynomial $f(x) = x^{256} \oplus x^{10} \oplus x^5 \oplus x^2 \oplus 1$.

Index i	Modifications	
	to plaintext M_{2i}, M_{2i+1} $(\mathbb{F}_{2^{256}})$	to ciphertext C'_{2i}, C'_{2i+1} (hex)
$i = 2$	$(2^4{+}1)\Delta = 2^{132} + 2^{128}$	$2^4\Delta = 00^{14}\|0010\|00^{16}$
$i = 3$	$(2^5{+}1)\Delta = 2^{133} + 2^{128}$	$2^5\Delta = 00^{14}\|0020\|00^{16}$
$i = 5$	$(2^7{+}1)\Delta = 2^{135} + 2^{128}$	$2^7\Delta = 00^{14}\|0080\|00^{16}$
$i = 8$	$(2^{10}{+}1)\Delta = 2^{138} + 2^{128}$	$2^{10}\Delta = 00^{14}\|0400\|00^{16}$
$i = 10$	$(2^{12}{+}1)\Delta = 2^{140} + 2^{128}$	$2^{12}\Delta = 00^{14}\|1000\|00^{16}$
$i = 254$	$(2^{256}{+}1)\Delta = 2^{138} + 2^{133} + 2^{130}$	$2^{256}\Delta = 00^{14}\|0425\|00^{16}$
$i = 256$	$(2^{258}{+}1)\Delta = 2^{140} + 2^{135} + 2^{132} + 2^{130} + 2^{128}$	$2^{258}\Delta = 00^{14}\|1094\|00^{16}$

For any example message M, we can now forge tag T^* and ciphertext C^* for the modified message M^*, which differs from M in blocks indices $j \in J$:

$$
\begin{aligned}
J &= \{4, 5, 6, 7, 10, 11, 16, 17, 20, 21, 508, 509, 512, 513\},\\
M^*_j &= \begin{cases} M_j \oplus (2^{\lfloor \frac{j}{2} \rfloor + 2}{+}1)\Delta & j \in J,\\ M_j & \text{else;} \end{cases}\\
C^*_j &= \begin{cases} C'_j \oplus 2^{\lfloor \frac{j}{2} \rfloor + 2}\Delta & j \in J,\\ C_j & \text{else;} \end{cases}\\
T^* &= T.
\end{aligned}
$$

This example can easily be verified with the reference implementation of Prøst-OTR-128 for any key K, nonce N and message M with ≥ 514 blocks, and the corresponding related values K', N' for $\Delta = 2^{128}$.

4 Remarks and advanced attacks

4.1 Remarks on the message length

If an attacker carries out the basic attack as in Section 3, the modified message may have a slightly modified bit length. This is because the modification can

shift the last nonzero bit, which marks the beginning of the message padding. This is not a problem since the message bitlength is not encoded anywhere else in the encryption process – except in the rare case that the last nonzero bit moves to the second-to-last block or earlier, which is not a valid format for the padded plaintext. This can be avoided by not including the last block pair in the modification process.

The attack is also applicable to messages $M = M_0 \cdots M_{2m-1} M_{2m}$ with an odd number of blocks: simply do not include the last block M_{2m} in the modification process, and copy it directly to M^*_{2m}. The same holds true for messages that include associated data A: simply copy the same associated data to the forged message.

4.2 Unknown messages

The description in Section 3 assumes that one and the same message M is encrypted under both keys, K and $K' = K \oplus \Delta$, and that M is known to the attacker. This is, however, not necessarily required. Even without knowing M, the attacker can compute forged ciphertext blocks and the tag. In this case, he will not know the modified message M^*, but only the induced difference $M^* \oplus M$.

Neither is it necessary that the same message M is encrypted under both K and $K \oplus \Delta$. In fact, it is sufficient that the attacker has access to the ciphertexts for any two (not necessarily known, not necessarily equal-length) messages M (under K) and M' (under $K' = K \oplus \Delta$), and knows the difference $M_{2i+1} \oplus M'_{2i+1}$ for at least $2n + 1$ values of i. Let I be the set of indices i with known message differences, with $|I| \geq 2n + 1$. Then, the attacker solves

$$\bigoplus_{i \in I} \lambda_i \left(M_{2i+1} \oplus M'_{2i+1} \oplus (2^{i+2} + 1)\Delta \right) = 0.$$

Again, a non-zero solution for λ exists since the $\geq 2n + 1$ vectors in $\mathbb{F}_2^{2n}$ must be linearly dependent.

The forged message M^* (not known to the attacker, same block length as M), ciphertext C^* and tag T^* are then given by

$$(M^*_{2i}, M^*_{2i+1}) = \begin{cases} (M_{2i}, M_{2i+1}) & i \notin I \vee \lambda_i = 0, \\ (M'_{2i} \oplus (2^{i+2} + 1)\Delta, M'_{2i+1} \oplus (2^{i+2} + 1)\Delta) & i \in I \wedge \lambda_i = 1; \end{cases}$$

$$(C^*_{2i}, C^*_{2i+1}) = \begin{cases} (C_{2i}, C_{2i+1}) & i \notin I \vee \lambda_i = 0, \\ (C'_{2i} \oplus 2^{i+2}\Delta, C'_{2i+1} \oplus 2^{i+2}\Delta) & i \in I \wedge \lambda_i = 1; \end{cases}$$

$$T^* = T.$$

4.3 Multiple forgeries

As described in Sections 3 and 4.2, an attacker can forge one message from $4n+2$ original message blocks. This can be extended to $2^s - 1$ different forgeries from

$4n + 2s$ blocks (i.e., $|I| \geq 2n + s$). Then, the homogenous linear system

$$\bigoplus_{i \in I} \lambda_i \left(M_{2i+1} \oplus M'_{2i+1} \oplus (2^{i+2} + 1)\Delta \right) = 0$$

is underdetermined with $\geq 2n + s$ variables for $2n$ equations. Thus, the solution space has dimension $\geq s$, containing $\geq 2^s - 1$ different non-zero solutions for λ.

In the case $M_j = M'_j$, different values λ, λ' produce different plaintexts as long as

$$\max\{i \in I : \lambda_i \neq \lambda'_i\} < \mathrm{ord}(2) - 2,$$

where $\mathrm{ord}(2)$ denotes the multiplicative order of 2 in $\mathbb{F}^*_{2^{2n}}$. For Prøst's irreducible polynomials, $\mathrm{ord}(2) = 2^{256} - 1$ for $n = 128$ and $\mathrm{ord}(2) = 2^{512} - 1$ for $n = 256$. In general, if

$$M_{2i+1} \oplus M'_{2i+1} \oplus (2^{i+2} + 1)\Delta \neq 0 \qquad \forall i \in I,$$

then all different λ produce different forgeries.

4.4 Almost universal forgery with related-key queries

Assume that the attacker can query for the encryption of a chosen message under one of the two keys, $K' = K \oplus \Delta$. He wants to forge the ciphertext and tag for a meaningful message M^* (chosen beforehand or provided externally) under the original key K. He can achieve this goal if (a) M^* has an even number of blocks, (b) he has access to the tag T of a known message M with the same number of blocks as M^* under the key K, and (c) he can modify one $2n$-bit block with odd index of M^* (or, alternatively, of M). The attack works as follows:

1. Fix the target message length $|M^*| = 2m$ (in blocks).
2. Obtain tag T for any known message M with $|M| = 2m$ under key K and any nonce N.
3. Fix the preliminary target (challenge) message M^*.
4. Let $j^* = 2i^* + 1$ be the modifiable block of M^*. Modify

$$M^*_{2i^*+1} = M_{2i^*+1} \oplus \bigoplus_{i \neq i^*} M_{2i+1} \oplus M^*_{2i+1}.$$

5. Construct the query message M' as

$$(M'_{2i}, M'_{2i+1}) = (M^*_{2i} \oplus (2^{i+1} \oplus 1)\Delta, M^*_{2i+1} \oplus (2^{i+1} \oplus 1)\Delta) \qquad i = 0, \ldots, m-1.$$

6. Request the ciphertext C' for the query message M' under $K' = K \oplus \Delta$ with nonce $N' \| 10^* = (N \| 10^*) \oplus \Delta$.
7. The forged ciphertext C^* and tag T^* for message M^* and nonce $N^* = N$ can be computed as

$$(C^*_{2i}, C^*_{2i+1}) = (C'_{2i} \oplus 2^{i+2}\Delta, C'_{2i+1} \oplus 2^{i+2}\Delta) \qquad i = 0, \ldots, m-1,$$
$$T^* = T.$$

This is essentially the same strategy as in Section 4.2, except that instead of using fixed M, M' and adapting M^*, we fix M, M^* and adapt M'. To avoid solving the equation system for the correct λ_i (which would require relatively long message lenghts $2m$, and force us to have $M^*_j = M_j$ for many j), we modify one block $M^*_{j^*}$ to make $\forall i : \lambda_i = 1$ a valid solution.

5 Discussion

The core of our attack is the following observation: If an authenticated encryption mode applies the block cipher to variable (controllable) inputs, an attacker can "lift" the inherent related-key weaknesses of the Even-Mansour construction to the entire mode. Then, he can use information from encryptions under a related key to forge ciphertext and tag for the target key.

A question that suggests itself is whether similar attacks are possible on other Prøst modes. In addition, other authenticated encryption modes might display similar problems when combined with an Even-Mansour block cipher.

Prøst-APE does not use the Even-Mansour construction at all, but plugs the permutation into a sponge construction. Thus, the attack is clearly not applicable. Prøst-COPA does use the permutation in an Even-Mansour construction. However, it seems to defy the attack by including $E_K(0)$, the encryption of the value 0, in the definition of the helper value L (which plays a role similar to ℓ in Prøst-OTR). Since a constant instead of the variable nonce N serves as input to the encryption, the input cannot be controlled to produce (differentially) predictable outputs of L. The situation is similar, for example, for the OCB mode of operation [17]: while the message could be used to cancel out differences in the helper counter value, this value is also derived from the encryption $E_K(0)$ of the zero value and thus unpredictable.

On the other hand, other popular modes show significant weaknesses when combined with Even-Mansour ciphers. Of course, unlike Prøst, these modes are usually not recommended for use with an Even-Mansour block cipher, but with AES. Consider, for example, the CCM mode of operation [10,24], an ISO/IEC-standardized combination of CBC-MAC with CTR encryption, as illustrated in Fig. 4. CCM allows a much simpler related-key attack. Assume that an attacker knows the ciphertext (including the tag) $C = C_1 \cdots C_\ell C_{\ell+1}$ of a message $M = M_1 \cdots M_\ell$ under key $K \oplus \Delta$ and padded nonce $(N\|0) \oplus \Delta$ (in the format used as counter input to the CTR encryption). Then, the ciphertext C' for M under key K and padded nonce $N\|0$ is simply

$$C'_i = \begin{cases} C_i \oplus \Delta & 1 \le i \le \ell, \\ C_i & i = \ell + 1. \end{cases}$$

As can be observed from Fig. 4, all differences Δ during the CCM computation cancel out either with the nonce difference fed to the Even-Mansour block encryptions $E_{K\oplus\Delta}$, or with neighbouring block cipher calls in the CBC-MAC computation. The final differences at the block cipher outputs from the CTR encryption can simply be added to the ciphertext blocks.

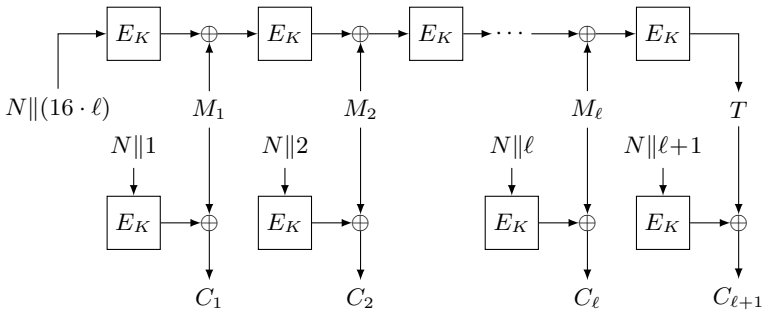


Fig. 4: CCM encryption.

Clearly, the Even-Mansour construction is not well-suited as a general-purpose block cipher construction for all modes of operation. The Prøst-OTR design is an example how even more complex modes can allow some undesirable properties of the Even-Mansour construction to be lifted to the complete authentication mode, in this case to generate related-key forgeries. The rising popularity of sponge modes and permutation-based encryption in general may lead to interesting new observations in this direction.

Finally, we stress again that the presented attack only concerns the OTR variant of Prøst. For this variant, powerful forgery attacks are possible in a related-key setting. The security of the other modes, Prøst-COPA and Prøst-APE, and in particular of the Prøst permutation itself, remains unaffected. It may be possible to tweak OTR to prevent the specific attack, for example by adapting the initialization of ℓ to include $\tilde{P}_K(0)$, similar to COPA and OCB. However, the general interactions of the OTR mode with the single-key Even-Mansour construction remains a reason for concern.

Acknowledgments. The work has been supported in part by the Austrian Science Fund (project P26494-N15) and by the Austrian Research Promotion Agency (FFG) and the Styrian Business Promotion Agency (SFG) under grant number 836628 (SeCoS).

References

1. Alomair, B.: AVALANCHE v1. Submission to the CAESAR competition: `http://competitions.cr.yp.to/caesar-submissions.html` (2014)
2. Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: APE: Authenticated permutation-based encryption for lightweight cryptography. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption – FSE 2014. LNCS, Springer (2014), in press, `http://eprint.iacr.org/2013/791`
3. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology – ASIACRYPT 2013. LNCS, vol. 8269, pp. 424–443. Springer (2013), `http://eprint.iacr.org/2013/790`

4. Biham, E.: New types of cryptoanalytic attacks using related keys (extended abstract). In: Helleseth, T. (ed.) Advances in Cryptology – EUROCRYPT '93. LNCS, vol. 765, pp. 398–409. Springer (1993)
5. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and related-key attack on the full AES-256. In: Halevi, S. (ed.) Advances in Cryptology – CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer (2009)
6. Biryukov, A., Wagner, D.: Advanced slide attacks. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000. LNCS, vol. 1807, pp. 589–606. Springer (2000)
7. Bogdanov, A., Knudsen, L.R., Leander, G., Standaert, F., Steinberger, J.P., Tischhauser, E.: Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations (extended abstract). In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. LNCS, vol. 7237, pp. 45–62. Springer (2012)
8. Daemen, J.: Limitations of the Even-Mansour construction. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) Advances in Cryptology - ASIACRYPT '91. LNCS, vol. 739, pp. 495–498. Springer (1991)
9. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in cryptography: The Even-Mansour scheme revisited. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. LNCS, vol. 7237, pp. 336–354. Springer (2012)
10. Dworkin, M.J.: SP 800-38C. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. Tech. rep., National Institute of Standards & Technology, Gaithersburg, MD, United States (2004)
11. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) Advances in Cryptology – ASIACRYPT '91. LNCS, vol. 739, pp. 210–224. Springer (1991)
12. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. Journal of Cryptology 10(3), 151–162 (1997)
13. Gentry, C., Ramzan, Z.: Eliminating random permutation oracles in the Even-Mansour cipher. In: Lee, P.J. (ed.) Advances in Cryptology – ASIACRYPT 2004. LNCS, vol. 3329, pp. 32–47. Springer (2004)
14. IEEE 802.11 working group: IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std 802.11-1997 (1997), `http://ieeexplore.ieee.org/servlet/opac?punumber=5258`
15. Kavun, E.B., Lauridsen, M.M., Leander, G., Rechberger, C., Schwabe, P., Yalçın, T.: Prøst v1. Submission to the CAESAR competition: `http://competitions.cr.yp.to/caesar-submissions.html` (2014)
16. Knudsen, L.R.: Cryptanalysis of LOKI. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) Advances in Cryptology – ASIACRYPT '91. LNCS, vol. 739, pp. 22–35. Springer (1991)
17. Krovetz, T., Rogaway, P.: The OCB authenticated-encryption algorithm. IETF RFC 7253: `http://tools.ietf.org/html/rfc7253` (2014)
18. Lampe, R., Patarin, J., Seurin, Y.: An asymptotically tight security analysis of the iterated Even-Mansour cipher. In: Wang, X., Sako, K. (eds.) Advances in Cryptology – ASIACRYPT 2012. LNCS, vol. 7658, pp. 278–295. Springer (2012)
19. Lee, Y., Jeong, K., Sung, J., Hong, S.: Related-key chosen IV attacks on Grain-v1 and Grain-128. In: Mu, Y., Susilo, W., Seberry, J. (eds.) Information Security and Privacy – ACISP 2008. LNCS, vol. 5107, pp. 321–335. Springer (2008)

20. Minematsu, K.: Parallelizable rate-1 authenticated encryption from pseudorandom functions. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology – EUROCRYPT 2014. LNCS, vol. 8441, pp. 275–292. Springer (2014)
21. Pasalic, E., Wei, Y.: Generic related-key and induced chosen IV attacks using the method of key differentiation. Cryptology ePrint Archive, Report 2013/586 (2013), http://eprint.iacr.org/2013/586
22. Taylor, C.: Calico v8. Submission to the CAESAR competition: http://competitions.cr.yp.to/caesar-submissions.html (2014)
23. The CAESAR committee: CAESAR: Competition for authenticated encryption: Security, applicability, and robustness (2014), http://competitions.cr.yp.to/caesar.html
24. Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM). IETF RFC 3610: http://tools.ietf.org/html/rfc3610 (2003)