Draft. Full paper not yet available.

The Associated-Data Problem

(How to cheaply authenticate unencrypted data when using an authenticated-encryption scheme)

Phillip Rogaway*

5 November 2001

Abstract

When using an authenticated-encryption scheme (a shared-key mechanism that provides both privacy and authenticity) it is sometimes useful, when encrypting a message, to also authenticate some additional information which is not privacy protected. We address this *associated-data problem*, wherein a Sender can bind to an authenticated ciphertext $\mathcal{C}$ a string *AD*, called its *associated-data*, and where the Receiver must provide the identical associated-data *AD* when processing $\mathcal{C}$—otherwise, the ciphertext will, almost certainly, be deemed *invalid*. We explain the utility of this problem, give a formal definition for it, and provide efficient solutions, both in general and for the authenticated-encryption scheme OCB.

Keywords: associated-data problem, authenticated encryption, block-cipher usage, cryptographic standards, modes of operation, OCB mode.

1 Introduction

The problem. During the last year and half there have emerged new block-cipher modes of operation which integrate privacy and authenticity protection in a single, compact mode. The first such scheme was suggested by Jutla [12], with Gligor et al. [9] and Rogaway et al. [18] soon offering related schemes. The new modes are an alternative to the generic composition approach, as named and analyzed by [4], where one glues together an arbitrary encryption scheme and an arbitrary MAC. Compared to doing that, the integrated modes promise several advantages, including improved efficiency and ease-of-correct-use. But the new modes would seem to have at least one disadvantage: an apparent inability to authenticate data without actually encrypting it and transmitting the associated ciphertext.

As an example showing where this can matter, consider a protocol that flows a message Msg = Header ‖ Ciphertext ‖ Tag, where Ciphertext is determined by encrypting some underlying Plaintext under a key K_{enc}, and Tag is determined by MACing Header ‖ Ciphertext under

*Department of Computer Science, University of California at Davis, Eng. II Building, Davis, California 95616 USA; and Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai 50200 Thailand. email: `rogaway@cs.ucdavis.edu` web: `http://www.cs.ucdavis.edu/~rogaway`

a key K_{mac}. Suppose we wish to modify this flow to employ an authenticated-encryption scheme such as OCB [18]. We can not just OCB-encrypt Header $\|$ Plaintext and send the resulting $\text{Ciphertext}_{\text{Header} \,\|\, \text{Plaintext}}$ in place of Msg because, presumably, Header had to be in the clear for purposes of routing or parsing the message. Nor can we OCB-encrypt Plaintext alone, sending the resulting $\text{Ciphertext}_{\text{Plaintext}}$ along with Header, for in this case we would have done nothing to authenticate Header. We could send $\text{Ciphertext}_{\text{Header} \,\|\, \text{Plaintext}}$ appended to Header, encrypting Header only to provide for its authenticity, but doing this would lengthen the message sent, which we certainly don't want.

In general, when a Receiver R gets an authenticated ciphertext $\mathcal{C}$ allegedly sent by a given Sender S, the Receiver R may wish to ensure not only that Sender S sent $\mathcal{C}$, but also that that Receiver R shares with Sender S a common understanding as to some further aspect of their current situation. This "further aspect of their situation" is encoded in a string AD, called the *associated-data.* The Sender and Receiver are expected to provide identical associated-data or else the Receiver should, almost certainly, detect the mismatch and reject the transmission $\mathcal{C}$.

In our earlier example, the associated-data would be Header. Here one anticipates that it is sent to the Receiver in the clear. In other examples, AD might represent information implicitly shared between the Sender and Receiver (eg., as a result of an earlier session-setup) such as cryptographic parameters that are in use, or the Sender's or Receiver's name or IP address. In cases like these the associated-data will be static over the course of the communication session. When this is the case, we aim for a solution that adds essentially zero per-message cost.

Origin of the problem. The associated-data problem was first described to the author by Burt Kaliski [13]. Shortly afterwards, it was independently suggested by Nancy Cam-Winget and Jesse Walker [7], and then by other individuals. Everyone who has asked about this problem has been involved in standardization efforts in which it became clear that one would sometimes like to bind to a ciphertext some additional, non-secret data. People wanted a cheap and secure way to do this when using an mechanism that integrates privacy and authenticity.

The naive solution, and doing better than it. As suggested earlier, one way to bind associated-data AD to an authenticated ciphertext is to have the Sender encode AD together with the plaintext M that he wants to communicate, and then encrypt-with-authenticity the resulting string $AD \,\|\, M$. The drawbacks of this are that it lengthens the ciphertext, and that it costs the Sender and Receiver additional computation time with every message that is sent, even when the associated-data AD is static during the entire session. We could try to erase the first inefficiency by having the Sender encrypt-with-authenticity $AD \,\|\, M$, and then drop from the resulting ciphertext that segment which corresponds to AD (assuming that the ciphertext has such a structure). But such an approach does not, in general, work; an authenticated-encryption mode may fail to provide authenticity if a portion of the ciphertext is not transmitted. Indeed modes like OCB [18] and IAPM [12] *do* fail to provide authenticity if ciphertext blocks are dropped. Here we seek a solution which applies to any authenticated-encryption scheme, rests on sound analysis, leaves fixed the length of a ciphertext, and adds essentially no overhead if AD is absent or static. Further efficiency goals, like avoiding the use of a new cryptographic key, will also be of interest.

Contributions. Our main contributions are as follows:

First, we give a definition for the security of an authenticated-encryption scheme allowing associated-data, an *AEAD-scheme.* The definition is very strong; in particular, the attack-model gives the adversary the ability to control AD, while the notion of adversarial success generalizes the notion of authenticity of ciphertexts [5, 14].

Second, we describe generic solutions to the associated-data problem. One technique, suggested by Cam-Winget and Walker [7], we call *nonce stealing*. The method is simple and useful, but somewhat limited in its applicability, as the string *AD* can only be a few bytes. A less restrictive approach, *ciphertext translation*, transforms an authenticated-encryption scheme that does not provide for associated-data (an AE-scheme) into an authenticated-encryption scheme that does (an AEAD-scheme). For this method one applies an xor-universal hash-function F to the string *AD*, and then xors the result Δ with a corresponding number of ciphertext bits, leaving the other bits alone: $\bar{\mathcal{E}}^{N,\,AD}_{KK'}(M) = \mathcal{E}^N_K(M) \oplus 0^*\Delta$, where $\Delta = F_{K'}(AD)$. Notice that if $F_{K'}(\varepsilon)$ is defined to be a string of 0-bits (where ε is the emptystring) then the constructed AEAD-scheme will be an extension of the original AE-scheme, in the sense that $\bar{\mathcal{E}}^{N,\,\varepsilon}_{KK'}(M) = \mathcal{E}^N_K(M)$. Also notice that if associated-data *AD* is held fixed during a communications session then the corresponding offset $\Delta = F_{K'}(AD)$ may be precomputed, essentially eliminating the per-message cost of authenticating *AD*. We prove that the constructed AEAD-scheme is secure as long as the AE-scheme one starts from is secure.

Third, we concretize and adjust the generic solution to yield a specific suggestion for OCB [18]. The solution retains OCB's use of a single block-cipher key. A nonempty message M is OCB-encrypted using a key K to get a ciphertext $\mathcal{C}$ which includes some τ bits of tag, and associated-data *AD* is PMAC-authenticated [19] under the same key K to get a τ-bit PRF-output of Δ. The final ciphertext is $\mathcal{C}$ with its last τ bits xored with Δ. Under this definition, $\mathrm{OCB}^{N,AD}_K(M)$ is fully parallelizable in both M and *AD*, and the function can be used as a pseudorandom function by fixing $M = \varepsilon$.

Comments. When using an xor-universal hash-function, the correctness of the AE⇒AEAD conversion relies on the AE-scheme meeting a strong definition of privacy: ciphertexts should be indistinguishable from random bits (when the adversary launches a chosen-plaintext attack), which we call IND\$-CPA. This is stronger than asking that ciphertexts be indistinguishable from the encryption of random bits, IND-CPA. The IND\$-CPA property is the one OCB was proven to achieve in [18]. The current note provides evidence that IND\$-CPA is a useful strengthening of IND-CPA. The IND\$-CPA property also allows the direct use of an encryption scheme as a pseudorandom generator or as a pseudorandom function from n-bit inputs to arbitrary-length outputs.

2 Preliminaries

AE-schemes. We follow [18] (which builds on [1, 5, 11]) in defining nonce-using authenticated-encryption schemes and their security. An *authenticated-encryption scheme* (an AE-scheme) is a three-tuple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. There are associated sets of strings $\mathsf{Nonce} = \{0,1\}^n$ and $\mathsf{Message}$, the latter having a simple (linear-time) membership test. The key space $\mathcal{K} = \{0,1\}^k$ is a nonempty set of strings. Algorithm $\mathcal{E}$ is a deterministic algorithm that takes strings $K \in \mathcal{K}$, $N \in \mathsf{Nonce}$, and $M \in \mathsf{Message}$, and returns a string $\mathcal{C} = \mathcal{E}^N_K(M) = \mathcal{E}_K(N, M)$. Algorithm $\mathcal{D}$ is a deterministic algorithm that takes strings $K \in \mathcal{K}$, $N \in \mathsf{Nonce}$, and $\mathcal{C} \in \{0,1\}^*$. The algorithm returns $\mathcal{D}^N_K(\mathcal{C})$, which is either a string in $\mathsf{Message}$ or the distinguished symbol Invalid. We require that $\mathcal{D}^N_K(\mathcal{E}^N_K(M)) = M$ for all $K \in \mathcal{K}$, $N \in \mathsf{Nonce}$, and $M \in \mathsf{Message}$.

An adversary with access to an oracle is *nonce-respecting* if the adversary never repeats the first argument to its oracle, regardless of oracle responses: if the adversary asks a query (N, M) then it never asks a subsequent query of (N, M').

Fix an AE-scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Let A be an algorithm having access to an oracle $\mathcal{E}_K(\cdot,\cdot)$, for a randomly chosen key K. We say that A *forges* if A is nonce-respecting and A outputs a triple $(N, \mathcal{C})$ where $\mathcal{D}^N_K(\mathcal{C}) \neq$ Invalid and A did *not* ask a query $\mathcal{E}^N_K(M)$ which resulted in a response $\mathcal{C}$. Let $\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(A)$ be the probability that A forges. Let $\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(q, \mu)$ denote the maximal

value of $\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(A)$ over all nonce-respecting adversaries that ask at most q queries, the sum of these queries, plus the length of the forgery attempt, being at most μ bits. Let $\mathbf{Adv}^{\mathrm{auth}}_{\Pi}(t, q, \mu)$ be identical, except that the adversary is also limited to running time plus description size of t, relative to some standard and fixed model of computation.

Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE-scheme for which the length of any ciphertext is a given function, $\ell(|M|)$, of the length of the plaintext M. Let $\$(\cdot,\cdot)$ be an oracle that, on input N, M, returns a random string of length $\ell(|M|)$. Let A be a adversary having access to an oracle. Let $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(A) = \Pr[K \xleftarrow{R} \mathcal{K} : A^{\mathcal{E}_K(\cdot,\cdot)} = 1] - \Pr[A^{\$(\cdot,\cdot)} = 1]$. We name this notion IND\$-CPA: indistinguishability from random bits under a chosen-plaintext attack. Let $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(q, \mu)$ denote the maximal value of $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(A)$ over all nonce-respecting adversaries A that ask at most q queries, the sum of the message-lengths in these queries being at most μ bits. Let $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(t, q, \mu)$ be identical, except that the adversary is also limited to running time plus description size of t.

Xor-universal hash functions. Function families and universality conditions on them originate with Carter and Wegman [8]. A function-family is a function $F : \mathcal{K} \times \mathcal{X} \to \{0,1\}^\tau$, where $\mathcal{K}$ has an associated distribution and $\mathcal{X} \subseteq \{0,1\}^*$. We consider the security property called *xor-universal*, first defined by [15]. For a function-family $F : \mathcal{K} \times \mathcal{X} \to \{0,1\}^\tau$, let $\mathbf{Adv}^{\mathrm{xu}}_F(\alpha) = \max_{x,x',c} \{\Pr[K \xleftarrow{R} \mathcal{K} : F_K(x) \oplus F_K(x') = c]\}$ where the maximum is over all $c \in \{0,1\}^\tau$ and over all distinct $x, x' \in \mathcal{X}$ subject to $|x|, |x'| \le \alpha$. For a complexity-theoretic analog, let $\mathbf{Adv}^{\mathrm{xu}}_F(t, \alpha) = \max_A\{\Pr[K \xleftarrow{R} \mathcal{K}; (x, x', c) \xleftarrow{R} A(\alpha) : F_K(x) \oplus F_K(x') = c]\}$ where the maximum is over all adversaries A that have running time plus description size of at most t and A outputs (x, x', c) such that $|x|, |x'| \le \alpha$ and $x = x'$.

Pseudorandom functions. The notion of a pseudorandom function originates with [10]; our treatment is a concrete-security one that follows [3]. Let $F : \mathcal{K} \times \mathcal{X} \to \{0,1\}^\tau$ be a function-family. Let $\mathrm{Rand}(\mathcal{X}, \tau)$ be the set of all functions from $\mathcal{X}$ to $\{0,1\}^\tau$. Then define $\mathbf{Adv}^{\mathrm{prf}}_F(A) = \Pr[K \xleftarrow{R} \mathcal{K} : A^{F_K(\cdot)} = 1] - \Pr[\rho \xleftarrow{R} \mathrm{Rand}(\mathcal{X}, \tau) : A^{\rho(\cdot)} = 1]$. Let $\mathbf{Adv}^{\mathrm{prf}}_F(q, \mu)$ be the maximal value of $\mathbf{Adv}^{\mathrm{prf}}_F(A)$ over all adversaries A that ask at most q oracle queries, these queries totaling at most μ bits. Let $\mathbf{Adv}^{\mathrm{prf}}_F(t, q, \mu)$ be identical except that A is also limited to running time plus description size of t. If $\mathcal{X} = \{0,1\}^n$ one omits the redundant resource parameter μ. Let $\mathrm{Perm}(n)$ be the set of all permutation from n bits to n bits. If $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ define $\mathbf{Adv}^{\mathrm{prp}}_F(A) = \Pr[K \xleftarrow{R} \mathcal{K} : A^{F_K(\cdot)} = 1] - \Pr[\pi \xleftarrow{R} \mathrm{Perm}(n) : A^{\pi(\cdot)} = 1]$. Let$\mathbf{Adv}^{\mathrm{prp}}_F(q)$ be the maximal value of $\mathbf{Adv}^{\mathrm{prp}}_F(A)$ among adversaries that ask q queries, and let $\mathbf{Adv}^{\mathrm{prp}}_F(t, q)$ be the value for adversaries also limited to running time plus description size of t.

Pseudorandom functions are xor-universal. The pseudorandom function requirement is stronger than the xor-universal one: for any function-family F with a τ-bit output, $\mathbf{Adv}^{\mathrm{xu}}_F(\alpha) \le \mathbf{Adv}^{\mathrm{prf}}(2, \alpha) + 2^{-\tau}$, since one possible statistical test is to ask oracle queries $F_K(x)$ and $F_K(x')$, for selected x, x', and test if the xor of these points is some particular value c. Similarly, $\mathbf{Adv}^{\mathrm{xu}}_F(t, \alpha) \le \mathbf{Adv}^{\mathrm{prf}}(t', 2, \alpha) + 2^{-\tau}$, where $t' = \Omega(t + \alpha + \tau)$.

3 Definition AEAD-Security

AEAD-schemes. An *authenticated-encryption scheme allowing associated-data* (henceforth an *AEAD-scheme*) is a three-tuple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. There are associated sets of strings $\mathsf{Nonce} = \{0,1\}^n$, Assoc, and $\mathsf{Message}$, the last two having a simple (linear-time) membership test. The key space $\mathcal{K}$ is a nonempty set of strings. Algorithm $\mathcal{E}$ is a deterministic algorithm that takes strings $K \in \mathcal{K}$,

$N \in \mathsf{Nonce}$, $AD \in \mathsf{Assoc}$, and $M \in \mathsf{Message}$. The algorithm returns a string $\mathcal{C} = \mathcal{E}_K^{N,AD}(M) = \mathcal{E}_K(N, AD, M)$. Algorithm $\mathcal{D}$ is a deterministic algorithm that takes strings $K \in \mathcal{K}$, $N \in \mathsf{Nonce}$, $AD \in \mathsf{Assoc}$, and $\mathcal{C} \in \{0,1\}^*$. The algorithm returns $\mathcal{D}_K^{N,AD}(\mathcal{C})$, which is either a string in $\mathsf{Message}$ or the distinguished symbol Invalid. We require that $\mathcal{D}_K^{N,AD}(\mathcal{E}_K^{N,AD}(M)) = M$ for all $K \in \mathcal{K}$, $N \in \mathsf{Nonce}$, $AD \in \mathsf{Assoc}$, and $M \in \mathsf{Message}$.

An adversary with access to an oracle is *nonce-respecting* if the adversary never repeats the first argument to its oracle, regardless of oracle responses: so if the adversary asks a query (N, AD, M) then it never asks a subsequent query of (N, AD', M').

Authenticity of an AEAD-scheme. Fix an AEAD-scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Let A be an algorithm having access to an oracle $\mathcal{E}_K(\cdot,\cdot,\cdot)$, for some randomly chosen key K. We say that A *forges* if A is nonce-respecting and A outputs a triple $(N, AD, \mathcal{C})$ where $\mathcal{D}_K^{N,AD}(\mathcal{C}) = \text{Invalid}$ and A did *not* ask a query $\mathcal{E}_K^{N,AD}(M)$ which resulted in a response $\mathcal{C}$. Let $\mathbf{Adv}_\Pi^{\text{AUTH}}(A)$ be the probability that A forges. Let $\mathbf{Adv}_\Pi^{\text{AUTH}}(q, \mu, \alpha)$ be the maximal value of $\mathbf{Adv}_\Pi^{\text{AUTH}}(A)$ over all nonce-respecting adversaries A that ask at most q queries, these, along with the forgery attempt and all associated data, totaling at most μ bits, and each associated-data string, including that in the forgery attempt, limited to α bits. Let $\mathbf{Adv}_\Pi^{\text{AUTH}}(t, q, \mu, \alpha)$ be identical, except that the adversary is also limited to running time plus description size of t.

Comments. The definition above is very strong—arguably stronger than what is "necessary" to capture the underlying intuition. In particular, the attack model is strong insofar as the adversary is allowed to manipulate both the nonce and the associated-data (subject to the constraint that no nonce is repeated), and the adversary's goal is modest insofar as she "gets credit" even for forgeries that use bizarre nonces and associated-data values, whether new or repetitions. In a real system, the message and the nonce will primarily be controlled by the Sender (for example, the nonce may be a counter) while the associated-data will primarily be chosen by the Sender and/or the Receiver. Still, an adversary may be able to influence these values. For example, an adversary might force a nonce to be incremented by thwarting a transmission from reaching its destination; or an adversary might induce the Sender to utilize bogus associated-data by manipulating flows in an unauthenticated handshake that proceeds the use of the AEAD-scheme. Allowing the adversary to manipulate all of M, N, and AD, and giving the adversary credit for any new $(N, AD, \mathcal{C})$, is a pessimistic approach that allows one to develop a robust definition.

The definition uses a space Assoc, with $AD \in \mathsf{Assoc}$, rather than referring to an arbitrary vector of strings $\boldsymbol{AD} \in (\{0,1\}^*)^*$, say. This turns out to be more convenient, and involves no real loss of generality: to allow vector-valued associated-data one has only to specify an injective and efficiently-computable encoding from $(\{0,1\}^*)^* \to \mathsf{Assoc}$, for some convenient set Assoc.

As usual, there is a certain degree of arbitrariness in how we have chosen to bound the resource parameters; the definitions of q, μ, α are made with an eye towards what our theorems will say.

Privacy of an AEAD-scheme. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AEAD-scheme. Assume that the length of any ciphertext is a given function, $\ell(|M|)$, of the length of the plaintext M. Let $\$(\cdot,\cdot,\cdot)$ be an oracle that, on input (N, M, AD), returns a random string of length $\ell(|M|)$. Let A be a adversary having access to an oracle. Let $\mathbf{Adv}_\Pi^{\text{PRIV}}(A) = \Pr[K \xleftarrow{R} \mathcal{K} : A^{\mathcal{E}_K(\cdot,\cdot)} = 1] - \Pr[A^{\$(\cdot,\cdot,\cdot)} = 1]$. Let $\mathbf{Adv}_\Pi^{\text{PRIV}}(q, \mu) = \Pr[K \xleftarrow{R} \mathcal{K} : A^{\mathcal{E}_K(\cdot,\cdot)} = 1] - \Pr[A^{\$(\cdot,\cdot,\cdot)} = 1]$ denote the maximal value of $\mathbf{Adv}_\Pi^{\text{PRIV}}(A)$ over all nonce-respecting adversaries A that ask at most q queries whose total length—message length plus associated-data length— totals at most μ bits. Let $\mathbf{Adv}_\Pi^{\text{PRIV}}(t, q, \mu)$ be identical, except that the adversary is also limited to running time plus description size of t.

4 Nonce Stealing

We now provide a first suggestion, due to Nancy Cam-Winget and Jesse Walker [7], for incorporating associated-data into an AE-scheme. We call the technique *nonce stealing*.

Suppose that the nonce in an AE-scheme is n bits, but that the application that uses this AE-scheme is content to use $n_0 < n$ bits for a nonce. For example, the nonce for the AE-scheme may be $n = 128$ bits, but an "application-layer nonce" of $n = 64$ bits may suffice for a given application. (A nonce of 32–64 bits would normally be adequate for applications that uses a counter for a nonce.) In such a case, associated-data may be packed into the "unused" $n - n_0$ bits of the AE-scheme's nonce.

At first glance, nonce stealing would sound to be of limited use, because so few bits of associated-data can be accommodated in this way. But often a few bytes is all that one needs, making the technique useful. The technique adds essentially no overhead, as well.

As an example, nonce stealing is anticipated for the IEEE 802.11 standard. The draft standard uses OCB-AES128, so the nonce is $n = 128$ bits. Of these 128 bits, the standard's designers currently anticipate that only 28 will be needed for the application nonce, which is a counter. The remaining 100 bits are associated-date: a source and destination address comprising 48 bits each, plus a quality-of-service indicator which is another 4 bits.

Nonce-AD is also anticipated for an IETF Internet Draft which will describe the use of OCB-AES128 as an IPsec transform [16]. This time the associated-data is the exact same data that is to be used as the application-level nonce: a 32-bit SPI and a 32-bit Sequence Number. In such a case, the way to authenticate the associated-data is to *do nothing*—we claim that the nonce of an AE-scheme is already automatically authenticated. We now justify this claim, thereby showing correctness for both forms of nonce stealing we have described.

Given AE-scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, define AEAD-scheme $\bar{\Pi} = (\mathcal{K}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ by setting $\bar{\mathcal{E}}_K^{N,N}(M) = \bar{\mathcal{E}}_K^N(M)$ and $\bar{\mathcal{D}}_K^{N,N}(\mathcal{C}) = \bar{\mathcal{D}}_K^N(\mathcal{C})$. Actually, this does not quite fit the syntax of an AEAD-scheme, since it does not allow arbitrary associated-data *AD* within some specified set of strings; instead, the associated-data must coincide with the nonce. Still, the definition of AEAD-authenticity makes sense even with this new restriction, so we continue undeterred. An adversary would be deemed successful in breaking the authenticity of the constructed AEAD-scheme if, after asking $\bar{\mathcal{E}}$-queries of $(N_1, N_1, M_1), \ldots, (N_q, N_q, M_q)$, getting responses $\mathcal{C}_1, \ldots, \mathcal{C}_q$, she produces a valid ciphertext $(N, N, \mathcal{C})$ where there was no earlier $\bar{\mathcal{E}}$-query (N, N, M_i) that resulted in a response $\mathcal{C}$. Thus, for the constructed AEAD-scheme, the adversary succeeds if, after asking $\mathcal{E}$-queries $(N_1, M_1), \ldots, (N_q, M_q)$, getting responses $\mathcal{C}_1, \ldots, \mathcal{C}_q$, she produces a valid ciphertext $(N, \mathcal{C})$ where there was no earlier $\mathcal{E}$-query (N, M_i) that resulted in a response $\mathcal{C}$. But this is precisely the definition for authenticity for an AE-scheme. In other words, under our definitions, security of nonce stealing is immediate.

The possibility of nonce stealing provides another reason, besides those enumerated in [18], why an AE-scheme is best designed to employ an arbitrary nonce. The possibility of nonce stealing also motivates the strong definition we have been using for authenticity of an AE-scheme: with a weaker definition of AE-scheme authenticity, nonce stealing likely would not work.

5 Ciphertext Translation

We now provide a solution to the AEAD-problem which permits arbitrary associated-data. The solution amounts to a method to transform an AE-scheme Π into an AEAD-scheme $\bar{\Pi}$ with the help of an xor-universal function-family F. We call the technique *ciphertext translation*.

First, a bit of notation. For $\mathcal{C}$ and Δ strings with $|\mathcal{C}| \geq |\Delta|$, let us write $\mathcal{C} \oplus 0^* \Delta$ for the

string $\mathcal{C} \oplus (0^{|\mathcal{C}|-|\Delta|} \parallel \Delta)$. (We are defining a binary operator named $\oplus 0^*$.) That is, $\mathcal{C} \oplus 0^* \Delta$ is $C \parallel T \oplus \Delta$, where $\mathcal{C} = CT$ and $|T| = |\Delta|$. For completeness, define $\mathcal{C} \oplus \Delta = \varepsilon$ if $|\mathcal{C}| < |\Delta|$.

Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE-scheme in which the length of any ciphertext is at least τ bits. Let $\mathsf{Assoc} \subseteq \{0,1\}^*$ be a given set of strings (with a linear-time membership test, let us say) and let $F : \mathcal{K}' \times \mathsf{Assoc} \to \{0,1\}^\tau$ be a function-family. Then we construct the AEAD-scheme $\bar{\Pi} = (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}}) = \Pi[F]$ as follows:

- The key space is $\bar{\mathcal{K}} = \mathcal{K} \times \mathcal{K}'$.
- Encryption is defined by $\bar{\mathcal{E}}_{KK'}^{N,\,AD}(M) = \mathcal{E}_K^N(M) \oplus 0^* F_{K'}(AD)$.
- Decryption is defined by $\bar{\mathcal{D}}_{KK'}^{N,\,AD}(\mathcal{C}) = \mathcal{D}_K^N(\mathcal{C} \oplus 0^* F_{K'}(AD))$.

That is, one takes the associated-data AD and computes from it $\Delta = F_{K'}(AD)$. Now, to encrypt, compute an authenticated ciphertext $\mathcal{C}$ without regards to AD, but then xor Δ with the last $|\Delta|$ bits of $\mathcal{C}$.

We comment that the security results that follow are indifferent to which bits of the ciphertext get modified by Δ.

Ciphertext-translation has the following pleasant properties: (1) the method extends any AE-scheme Π, and without regards to the internal structure of Π; (2) the method is parameterized by an arbitrary function-family F, with the requisite property of F soon to be explained; (3) when AD is static over the course of a session (or even over the course of several messages), the value $\Delta = F_{K'}(AD)$ may be profitably precomputed; (4) the approach adds essentially no overhead to an AE-scheme when associated-data is *not* used; (5) the method gives rise to a proper extension of the AE-scheme, $\bar{\mathcal{E}}_{KK'}^{N,\,\varepsilon}(M) = \mathcal{E}_K^N(M)$, as long as one sees to it that $F_{K'}(\varepsilon) = 0^\tau$.

Ciphertext-translation has the following unpleasant property: (a) it uses a new key, K', different from the keys used for Π. This disadvantage will be addressed in Section 9.

6 Security of Ciphertext Translation

We now show the following theorem on the security of ciphertext translation.

Theorem 1 *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE-scheme in which each ciphertext has at least τ bits, and let $F : \mathcal{K}' \times \mathsf{Assoc} \to \{0,1\}^\tau$ be a function-family. Then*

$$\begin{aligned} \mathbf{Adv}_{\Pi[F]}^{\mathrm{AUTH}}(q,\mu,\alpha) &\le \mathbf{Adv}_{\Pi}^{\mathrm{auth}}(q,\mu) + \mathbf{Adv}_{\Pi}^{\mathrm{priv}}(q,\mu) + \mathbf{Adv}_{F}^{\mathrm{xu}}(\alpha) \\ \mathbf{Adv}_{\Pi[F]}^{\mathrm{PRIV}}(q,\mu) &\le \mathbf{Adv}_{\Pi}^{\mathrm{priv}}(q,\mu)\ . \end{aligned}$$

Proof: We begin by proving the authenticity claim. Let A be an adversary that attacks the authenticity of $\Pi[F] = (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ using resources (q, μ, α). We construct an adversary B that attacks the authenticity of Π, uses resources (q, μ), and achieves advantage $\mathbf{Adv}_{\Pi}^{\mathrm{auth}}(B) \ge \mathbf{Adv}_{\Pi}^{\mathrm{AUTH}}(A) - \mathbf{Adv}_{\Pi}^{\mathrm{priv}}(q,\mu) - \mathbf{Adv}_{F}^{\mathrm{xu}}(\alpha)$.

Definition of the adversary B. Adversary B works as follows. First, B chooses a random $K' \xleftarrow{R} \mathcal{K}'$. Then B runs A. When A makes its ith oracle query, (N_i, AD_i, M_i), adversary B makes the query of (N_i, M_i) to it own oracle. Adversary B receives a response $\mathcal{C}_i = C_i \parallel T_i$, computes $\Delta_i = F_{K'}(AD_i)$, and provides to A the ciphertext $\mathcal{C}_i \oplus 0^* \Delta_i$. After A makes its q oracle queries (and B makes the correspond q oracle queries), adversary A makes a forgery attempt $(N, AD, \mathcal{C})$, where $\mathcal{C} = C \parallel T$. At that point adversary B computes $\Delta = F_{K'}(AD)$ and makes its own forgery attempt of $(N, \mathcal{C} \oplus 0^* \Delta)$. Clearly B uses the claimed resources (q, μ).

Analysis of B's forgery probability. Let predicate $\mathsf{Aforges}(a, K', K)$ be `true` iff A forges when A's internal coins are a and A interacts with an encryption oracle $\bar{\mathcal{E}}$ that uses coins (K, K'). Let predicate $\mathsf{Bforges}(a, K', K)$ be `true` iff B forges when B's internal coins are (a, K') and B interacts with an encryption oracle $\mathcal{E}$ that uses internal coins of K.

We claim that for "most" values of (a, K', K), we have that $\mathsf{Aforges}(a, K', K) = \mathsf{Bforges}(a, K', K)$. Indeed a case analysis shows that the only time when $\mathsf{Aforges}(a, K', K)$ could be different from $\mathsf{Bforges}(a, K', K)$ is when (a, K', K) results in A making a forgery attempt $(N_i,\ AD,\ C_i \parallel T_i)$ where $AD = AD_i$. In that case it is possible that $\mathsf{Aforges}(a, K', K) = \texttt{true}$ but $\mathsf{Bforges}(a, K', K) = \texttt{false}$. The case analysis is as follows. Run A in the manner determined by (a, K', K), defining the variables (N_i, AD_i, M_i), Δ_i, $\mathcal{C}_i = C_i \parallel T_i$, and $(N, AD, \mathcal{C})$, where $\mathcal{C} = C \parallel T$. There is a corresponding execution of B with identical associated variables. One checks that, in the two executions,

- if $N \notin \{N_1, N_2, \ldots, N_q\}$ then $A(a, K', K)$ forges iff $B(a, K', K)$ forges.
- if $N = N_i$ (for some i) and $C = C_i$, then $A(a, K', K)$ forges iff $B(a, K', K)$ forges.
- if $N = N_i$ (for some i) and $C = C_i$ and $AD = AD_i$ then $A(a, K', K)$ forges iff $B(a, K', K)$ forges.

We are left to bound the probability, over (a, K', K), that $N = N_i$ (for some i), $C = C_i$, $AD = AD_i$, $\mathsf{Aforges}(a, K', K) = \texttt{true}$, and $\mathsf{Bforges}(a, K', K) = \texttt{false}$. This event happens iff adversary A forges with $(N, AD, \mathcal{C})$, where $\mathcal{C} = C \parallel T$, after having asked an earlier query (N, AD_i, M_i) that resulted in ciphertext $\mathcal{C}_i = C \parallel T_i$, and $T_i \oplus F_{K'}(AD_i) = T \oplus F_{K'}(AD)$. In such a case, the forgery attempt can be valid for A (as AD is new) but invalid for B (the forgery attempt being a repetition). We wish to show that this case rarely occurs: over random (a, K', K), almost certainly $T_i \oplus F_{K'}(AD_i) = T \oplus F_{K'}(AD)$. Now $\Pr[T_i \oplus F_{K'}(AD_i) = T \oplus F_{K'}(AD)] = \Pr[F_{K'}(AD_i) \oplus F_{K'}(AD) = c]$, where $c = T_i \oplus T$, and we would like to conclude that this value is at most $\mathbf{Adv}^{\mathrm{xu}}_F(\alpha)$ by the definition of xor-universality. However, this isn't quite true: were adversary A's the encryption oracle to leak *no* information about K' to A, then A could find c and distinct AD, AD_i of length at most α such that $F_{K'}(AD_i) \oplus F_{K'}(AD) = c$ with probability, over K', of at most $\mathbf{Adv}^{\mathrm{xu}}_F(\alpha)$. However, the encryption scheme Π is not perfectly private.

To deal with this, let

$$\delta \;=\; \Pr[A\text{'s attack yields } T, T_i,\ AD_i = AD \text{ s.t. } T_i \oplus F_{K'}(AD_i) = T \oplus F_{K'}(AD)] - \mathbf{Adv}^{\mathrm{xu}}_F(\alpha)$$

and note that there is an adversary D that distinguishes $\mathcal{E}$-encrypted text from random bits that achieves advantage δ and that asks at most q queries and μ total bits. The adversary D behaves like the adversary B we have defined, but instead of outputting a forged ciphertext the adversary computes whether or not, for B's forged ciphertext and earlier queries, $C = C_i$ and $T_i \oplus F_{K'}(AD_i) = T \oplus F_{K'}(AD)$. If this inequality holds, D outputs 1; otherwise, D outputs 0. The adversary achieves advantage δ and runs with resources (q, μ). We conclude that $\delta \le \mathbf{Adv}^{\mathrm{priv}}_{\Pi}(q, \mu)$. We conclude that $\mathbf{Adv}^{\mathrm{AUTH}}_{\Pi[F]}(q, \mu, \alpha) \le \mathbf{Adv}^{\mathrm{auth}}_{\Pi}(q, \mu) + \mathbf{Adv}^{\mathrm{xu}}_F(\alpha) + \delta \le \mathbf{Adv}^{\mathrm{AUTH}}_{\Pi[F]}(q, \mu, \alpha) \le \mathbf{Adv}^{\mathrm{auth}}_{\Pi}(q, \mu) + \mathbf{Adv}^{\mathrm{priv}}_{\Pi}(q, \mu) + \mathbf{Adv}^{\mathrm{xu}}_F(\alpha)$, finishing the authenticity claim in the theorem.

Privacy. The second inequality in the theorem statement is easy. For convenience, we reuse the names A and B. Let A be an adversary that attacks the privacy of $\Pi[F] = (\bar{\mathcal{K}}, \bar{\mathcal{E}}, \bar{\mathcal{D}})$ using resources (q, μ, α). We construct an adversary B that attacks the privacy of Π, uses resources (q, μ), and achieves advantage $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(B) \ge \mathbf{Adv}^{\mathrm{PRIV}}_{\Pi}(A)$. Adversary B works as follows. First, B chooses a random $K' \xleftarrow{R} \mathcal{K}'$. Then B runs A. When A makes its ith oracle query, (N_i, AD_i, M_i), adversary B

makes the query of (N_i, M_i) to it own oracle. Adversary B receives a response $\mathcal{C}_i = C_i \parallel T_i$, computes $\Delta_i = F_{K'}(AD_i)$, and provides to A the ciphertext $\mathcal{C}_i \oplus 0^* \ \Delta_i$. After A makes its q oracle queries (and B makes the correspond q oracle queries), adversary A outputs a bit b. At that point adversary B outputs the same bit b. Clearly B uses the claimed resources (q, μ), and, since B perfectly simulates the native environment for A, $\mathbf{Adv}^{\mathrm{PRIV}}_{\Pi[F]}(A) = \mathbf{Adv}^{\mathrm{priv}}_{\Pi}(B)$, completing the theorem. ▮

As usual, there is a complexity-theoretic analog to Theorem 1. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE-scheme in which each ciphertext has at least τ bits, and let $F : \mathcal{K}' \times \mathsf{Assoc} \to \{0,1\}^\tau$ be a function-family. Then $\mathbf{Adv}^{\mathrm{AUTH}}_{\Pi[F]}(t, q, \mu, \alpha) \leq \mathbf{Adv}^{\mathrm{auth}}_{\Pi}(t', q, \mu) + \mathbf{Adv}^{\mathrm{priv}}_{\Pi}(t', q, \mu) + \mathbf{Adv}^{\mathrm{xu}}_{F}(t', \alpha)$ and $\mathbf{Adv}^{\mathrm{PRIV}}_{\Pi[F]}(t, q, \mu) \leq \mathbf{Adv}^{\mathrm{priv}}_{\Pi}(t', q, \mu)$ where $t' = t + O(\mu) + \mathrm{Time}_F(q, \mu)$, where $\mathrm{Time}_F(q, \mu)$ denotes the time to compute $K' \xleftarrow{R} \mathcal{K}'$ plus the time to compute $F_{K'}$ on at most $q + 2$ points, these points totaling at most μ bits.

7 Using a PRF instead of an XOR-Universal Function-Family

While the function-family F used by ciphertext translation needs only to be good in the sense of being xor-universal, there are a couple of advantages to selecting a function-family that meets the stronger notion of being a good PRF. One advantage of using a PRF for F is that it allows one to use an AE-scheme that meets a weaker notion of privacy, namely, IND-CPA. But this advantage may be of no practical significance, since all the proposed authenticated-encryption modes seem to meet the stronger IND\$-CPA notion, anyway.

A more significant advantage of using a good PRF is that it facilitates using the AEAD-scheme as a MAC. This addresses a question of Ron Rivest [17], who asked if OCB can be used in some simple manner to give a MAC, or to give other useful tools. Note that trying to use OCB or IAPM as a MAC by sending only the tag block does *not* work. But if one uses the ciphertext translation construction with a pseudorandom function, one gets a good MAC in a different way: just regard the message to encrypt as the empty string, and regarding the message to MAC as the associated-data. In this way one actually gets something better than a MAC: there is now a pseudorandom function embedded within, and still accessible though, the AEAD-interface. In particular, set the nonce to $N = \mathbf{0}$, say, regard the message M which one wants to MAC or as associated-data, and then encrypt the emptystring. The scheme is deterministic and stateless: one can keep reusing the nonce $N = \mathbf{0}$ in this way, even though it is not normally acceptable to repeat a nonce.

In order to ensure that the AEAD-scheme is an extension of the underlying AE-scheme, that is, that $\bar{\mathcal{E}}^{N,\varepsilon}_{KK'}(M) = \mathcal{E}^N_K(M)$, it is desirable to arrange that $F_{K'}(\varepsilon) = 0^\tau$. Of course hardwiring $F_{K'}(\varepsilon) = 0^\tau$ keeps F from being a PRF over the entire domain of F: one can distinguish $g = F_{K'}$ from a random function g by asking for g at ε. But when F is used within the ciphertext-translation construction with an AE-scheme meeting IND\$-CPA security, $\bar{\mathcal{E}}^{\mathbf{0},AD}_{KK'}(\varepsilon)$ should again give a good PRF over the entire domain Assoc. In any case, being a good PRF across nonempty strings should be quite good enough for applications, so we do not pursue the question further.

8 Instantiating Function-Family F

In this section we give some suggested functions F to use within the ciphertext-translation construction: CBCMAC$^\tau$, XORMAC$^\tau$, or PMAC$^\tau$. All of these functions are built from a block cipher $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and target complexity-theoretic statements. Entirely information-theoretic constructions, and constructions which do not use a block cipher, are certainly possible.

Proofs of the relevant bounds are omitted from this draft.

Using CBCMAC. Let $\mathsf{Assoc} = (\{0,1\}^n)^*$ and let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Define $\mathrm{CBCMAC}_a^\tau(\varepsilon) = 0^\tau$ and, for $x \in (\{0,1\}^n)^+$, and define $\mathrm{CBCMAC}_a^\tau(x_1 \cdots x_m)$, where $|x_1| = \cdots = |x_m| = n$, as the first τ bits of y_m, where $y_0 = 0^n$ and $y_i = E_a(y_{i-1} \oplus x_i)$. Though CBCMAC^τ is not a PRF on the set Assoc, it is computationally xor-universal, with good bounds, on this set. This can be shown by adapting the proofs in [6], say.

Using the XORMAC core. Fix $\ell \in [1..n-1]$ (e.g., $\ell = 8$ or $\ell = 32$) and let $\mathsf{Assoc} = (\{0,1\}^{n-\ell})^{<2^\ell}$. For $i \in [1..2^i - 1]$, let $\langle i \rangle$ denote the encoding of i into ℓ bits. Let $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Define $\mathrm{XORMAC}_a^\tau(\varepsilon) = 0^\tau$ and, for $x = x_1 \ldots x_m$, $|x_1| = \ldots = |x_m| = n-\ell$, define $\mathrm{XORMAC}_a^\tau(x)$ as the first τ bits of $E_a(\langle 1 \rangle \parallel x_1) \oplus E_a(\langle 2 \rangle \parallel x_2) \oplus \cdots \oplus E_a(\langle m \rangle \parallel x_m)$. Though XORMAC^τ is not a PRF on the set Assoc, it is computationally xor-universal on this set. This function-family is the core of the XOR MAC [2]. The function is computationally xor-universal, with good bounds, on the set Assoc. Note that this function is fully parallelizable.

Using PMAC. Let $\mathsf{Assoc} = \{0,1\}^*$. Define $\mathrm{PMAC}_a^\tau(\varepsilon) = 0^\tau$, and, for all other values of x, define $\mathrm{PMAC}_a^\tau(x) = \mathrm{PMAC}[E,\tau]\,(x)$, as specified in [19]. By the results in [19], this function is a PRF on the set $\{0,1\}^+$. This function is fully parallelizable, approximately as fast to compute as the CBC MAC, and is defined on all bit strings.

9 Avoiding Multiple Keys when Using OCB

According to what has been said so far, one needs to use two different keys to solve the associated-data problem using ciphertext translation: we set $\bar{\mathcal{E}}_{KK'}^{N,\,AD}(M) = \mathcal{E}_K^N(M) \oplus 0^*\ F_{K'}(AD)$, and the "key-reusing definition" of $\bar{\mathcal{E}}_K^{N,\,AD}(M) = \mathcal{E}_K^N(M) \oplus 0^*\ F_K(AD)$ certainly will not, in general, work. Nonetheless, we single out a case where the key-reusing definition $\bar{\mathcal{E}}_K^{N,\,AD}(M) = \mathcal{E}_K^N(M) \oplus 0^*\ F_K(AD)$ *does* work: Couple OCB with PMAC, encrypting by $\mathrm{OCB}_K^{N,\,AD}(M) = \mathrm{OCB}_K^N(M) \oplus 0^*\ \mathrm{PMAC}_K^\tau(AD)$.

To prove the security of this scheme one needs to establish that there is no bad "interference" between the two schemes. The proof is non-trivial because there is "interaction" between OCB and PMAC when keyed by the same key: beyond the common definition of "L" in the two schemes, which is easily dealt with, OCB defines $R = E_K(N \oplus L)$ and PMAC defines $C[1] = E_K(M[1] \oplus L)$. In the formal model, both N and $M[1]$ are under the adversary's control. This type of interaction between the two schemes would seem to spell trouble. All the same, one can prove that PMAC_K remains a good pseudorandom function even in the presence of an oracle for OCB-encryption and ciphertext-validity verification, these under the same key K that keys PMAC. This fact can be used to justify key-reuse across these two functions. Details are postponed until the full paper.

Acknowledgments

Burt Kaliski first suggested the AEAD problem to me. Mihir Bellare, John Black, Nancy Cam-Winget, Burt Kaliski, Robert Moskowitz, Ron Rivest, and Jesse Walker all gave useful comments and information. Work on the associated-data problem was furthered because of NIST's modes-of-operation effort; thanks to Elaine Barker, William Burr, Morris Dworkin, and the others at NIST who have been involved. The current note is responsive to a promise the author made for a writeup during NIST's second Mode of Operation workshop (Aug 24, 2001).

This work was funded under NSF CCR-9625460 and by a gift from CISCO Systems. Special thanks to CISCO's Dave McGrew, who has followed and championed my work.

This work was carried out while the author was at Department of Computer Science, Faculty of Science, Chiang Mai University, Thailand. Many thanks to CMU for extending, as usual, their gracious hospitality.

References

[1] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. *Proceedings of 38th Annual Symposium on Foundations of Computer Science* (FOCS 97), IEEE, 1997. `http://www.cs.ucdavis.edu/~rogaway/`

[2] M. Bellare, R. Guérin, and P. Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. *Advances in Cryptology – CRYPTO '95.* Lecture Notes in Computer Science, vol. 963, Springer-Verlag, D. Coppersmith, ed., pp. 15–28, 1995. `http://www.cs.ucdavis.edu/~rogaway/`

[3] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, vol. 61, no. 3, Dec 2000. (Earlier version in *Advances in Cryptology – CRYPTO '94.*) `http://www.cs.ucdavis.edu/~rogaway/`)

[4] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Advances in Cryptology – ASIACRYPT '00.* Lecture Notes in Computer Science, vol. 1976, T. Okamoto., ed., Springer-Verlag, 2000. `http://www-cse.ucsd.edu/users/mihir/`

[5] M. Bellare and P. Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient encryption. *Advances in Cryptology – ASIACRYPT '00.* Lecture Notes in Computer Science, vol. 1976, T. Okamoto., ed., Springer-Verlag, 2000. `http://www.cs.ucdavis.edu/~rogaway/`

[6] J. Black and P. Rogaway. CBC MACs for arbitrary-length messages: The three-key constructions. Lecture Notes in Computer Science, vol. 1880, pp. 187–215, 2000. `http://www.cs.ucdavis.edu/~rogaway`

[7] N. Cam-Winget and J. Walker. Personal communications, June 2001.

[8] L. Carter and M. Wegman. Universal hash functions. *J. of Computer and System Sciences*, vol. 18, pp. 143–154, 1979.

[9] V. Gligor and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. *Fast Software Encryption*, Lecture Notes in Computer Science, Springer-Verlag, April 2001.

[10] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM,* vol. 33, no. 4, pp. 210–217, 1986.

[11] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, vol. 28, April 1984, pp. 270–299.

[12] C. Jutla. Encryption modes with almost free message integrity. *Advances in Cryptology – EUROCRYPT 2001.* Lecture Notes in Computer Science, vol. 2045, B. Pfitzmann, ed., Springer-Verlag, 2001.

[13] B. Kaliski. Personal communication, May 2001.

[14] J. Katz and M. Yung. Unforgeable encryption and adaptively secure modes of operation. *Fast Software Encryption '00*. Lecture Notes in Computer Science, B. Schneier, ed., 2000.

[15] H. Krawczyk. LFSR-based hashing and authentication. *Advances in Cryptology — CRYPTO '94*. Lecture Notes in Computer Science, vol. 839, Springe-Verlag, pp. 129–139, 1994.

[16] R. Moskowitz. Personal communications. July 2001.

[17] R. Rivest. Personal communications, Aug 2001.

[18] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. *Eighth ACM Conference on Computer and Communications Security* (CCS-8). ACM Press, 2001. `http://www.cs.ucdavis.edu/~rogaway`

[19] P. Rogaway and J. Black. PMAC. Cryptology ePrint archive, report 2001/27, April 1, 2001 (revised April 18, 2001). `http://www.cs.ucdavis.edu/~rogaway`

[20] G. Rose. E-mail comments to NIST (idea attributed to P. Hawkes and G. Rose) with subject heading "Mode of Operations combining authentication and partial encryption." Aug 29, 2001.

A Other Proposals

Phil Hawkes and Greg Rose [20] suggest to address the associated-data problem in Jutla's IAPM by first modifying the definition of the checksum to be the xor of all plaintext blocks *and all ciphertext blocks*; then placing the associated-data in any understood locations of the plaintext, as long as it falls along block boundaries; and then omitting from the transmitted ciphertext all ciphertext blocks which correspond to plaintext blocks of associated-data. This suggestion may work—it seems plausible. However: (i) there are no definitions of the goal and no proofs; (ii) the change is specific to IAPM; (iii) the approach adds computational overhead (for the new xors) regardless of whether or not one has associated-data; and (iv) the approach adds computational overhead (for encrypting the *AD* blocks) even when the associated-data is static during the course of a communications session.