IEEE STANDARDS ASSOCIATION

IEEE Standard for Wide-Block Encryption for Shared Storage Media

IEEE Computer Society

Sponsored by the
Information Assurance Standards Committee
and
Storage Systems Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1619.2™-2010

8 March 2011

IEEE Standard for Wide-Block Encryption for Shared Storage Media

Sponsor

Information Assurance Standards Committee
and
Storage Systems Standards Committee
of the
IEEE Computer Society

Approved 30 September 2010

IEEE-SA Standards Board

Approved 5 May 2011

American National Standards Institute

Abstract: EME2-AES and XCB-AES wide-block encryption with associated data (EAD) modes of the NIST AES block cipher, providing usage guidelines and test vectors, are described. A wide-block encryption algorithm behaves as a single block cipher with a large plaintext input and ciphertext output, but uses a narrow block cipher [in this case Advanced Encryption Standard (AES)] internally. These encryption modes are oriented toward random access storage devices that do not provide authentication, but need to reduce the granularity of a potential attack.
Keywords: data-at-rest security, encryption, encryption with associated data (EAD), encrypt-mix-encrypt-v2 mode of operation (EME2), extended codebook mode of operation (XCB), IEEE 1619.2, security, storage

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-7381-6476-2 STD97026
Print: ISBN 978-0-7381-6477-9 STDPD97026

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Introduction

This introduction is not part of IEEE Std 1619.2-2010, IEEE Standard for Wide-Block Encryption for Shared Storage Media.

The purpose of this standard, similar to IEEE Std 1619-2007 [B2], is to describe a method of encryption for data stored in logical block-based devices, where the threat model includes possible access to stored data by the adversary.[a] As in IEEE Std 1619-2007, this standard specifies length-preserving encryption algorithms to be applied to the plaintext logical block before storing it on the storage media.

This standard improves on IEEE Std 1619-2007 by defining *wide-block* encryption algorithms. This means that they act on the whole logical block at once, and each bit on the input plaintext influences every bit of the output ciphertext (and vice versa for decryption). In particular, this standard specifies the EME2-AES and the XCB-AES wide-block encryption algorithms.

Wide-block encryption better hides plaintext statistics and provides better protection than the narrow-block encryption, defined in IEEE Std 1619-2007, against attacks that involve traffic analysis and/or manipulations of ciphertext on the raw storage media.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

[a] The numbers in brackets correspond to those of the bibliography in Annex A.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. A patent holder or patent applicant has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses. Other Essential Patent Claims may exist for which a statement of assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the Security in Storage Working Group had the following sponsorship:

James P. Hughes, *Sponsor Chair (IASC)*
Eric A. Hibbard, *Sponsor Vice Chair (IASC)*
John L. Cole, *Past Sponsor Chair (IASC)*
Curtis Anderson, *Co-Sponsor Chair (SSSC)*

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the Security in Storage Working Group had the following membership:

Matthew V. Ball, *Chair*
Eric A. Hibbard, *Vice Chair*
Walter Hubis, *Secretary*
Fabio Maino, *Technical Editor and Past Secretary*
James P. Hughes, *Past Chair*

Gideon Avida
Jim Coomes
Robert Elliott
Hal Finney
John Geldman
Bob Griffin
Cyril Guyot
Shai Halevi
Laszlo Hars
Larry Hofer
Glen Jaquette
Scott Kipp
Curt Kolovson
Robert Lockhart
Charlie Martin
David McGrew
Gary Moorhead
Bob Nixon
Landon Curt Noll
Jim Norton
Scott Painter
Dave Peterson
Serge Plotkin
Niels Reimers
Subhash Sankuratripati
David Sheehy
Bob Snively
Joel Spencer
Doug Whiting
Mike Witkowski

Special thanks for their important technical contribution to this standard to the following individuals:

Hal Finney
Brian Gladman
Shai Halevi
David McGrew

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Johann Amsenga
Khin Mi Mi Aung
Matthew V. Ball
Rahul Bhushan
Juan Carreon
Keith Chow
John Cole
Geoffrey Darnton
Russell Dietz
Thomas Dineen
Robert Elliott
Andrew Fieldsend
C. Fitzgerald
John Geldman
Ron Greenthaler
Randall Groves
Laszlo Hars
Eric A. Hibbard
Werner Hoelzl
Larry Hofer
Walter Hubis
Raj Jain
Scott Kipp
Susan Land
Kenneth Lang
Daniel Levesque
Robert Lockhart
William Lumpkins
G. Luri
Fabio Maino
Edward McCall
Jeffrey Moore
Finnbarr Murphy
Michael S. Newman
Landon Curt Noll
Ulrich Pohl
Randall Safier
Bartien Sayogo
Stephen Schwarm
Akihiro Shimura
Gil Shultz
Steven Smith
Kapil Sood
Thomas Starai
Rene Struik
Walter Struppler
Joseph Tardo
Brian Weis
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 30 September 2010, it had the following membership:

Robert M. Grow, *Chair*
Richard H. Hulett, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Karen Bartleson
Victor Berman
Ted Burse
Clint Chaplin
Andy Drozd
Alexander Gelman
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Hung Ling
Oleg Logvinov
Ted Olsen
Ronald C. Petersen
Thomas Prevost
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Michelle Turner
IEEE Standards Program Manager, Document Development

Michael D. Kipness
IEEE Standards Program Manager, Technical Program Development

Contents

IEEE Standard for Wide-Block Encryption for Shared Storage Media

1. Overview

1.1 Scope

This standard specifies an architecture for encryption of data in random access storage devices, oriented toward applications that benefit from wide encryption-block sizes of 512 bytes and above.

1.2 Purpose

This standard specifies an architecture for media security and enabling components. Wide encryption blocks are well suited to environments where the attacker has repeated access to cryptographic communication or ciphertext, or is able to perform traffic analysis of data access patterns. The standard is oriented toward fixed-size encryption blocks without data expansion, but anticipates an optional data expansion mode to resist attacks involving data tampering.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

NIST Federal Information Processing Standard 197 (FIPS 197), Advanced Encryption Standard (AES). November 2001.[1]

NIST Special Publication 800-38A (NIST SP 800-38A), Recommendation for Block Cipher Modes of Operation—Methods and Techniques.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. *The IEEE Standards Dictionary: Glossary of Terms & Definitions* should be referenced for terms not defined in this clause.[2]

associated data: Data that is associated with the plaintext, but that does not need to be encrypted. The associated data input should characterize the plaintext. In other algorithms for encryption of data at rest, the associated data is often referred to as *tweak value*.

encryption with associated data (EAD): A cryptographic algorithm that consists of an encryption procedure used to encrypt a plaintext and the associated data and a secret key, and a decryption procedure used to decrypt a ciphertext with the same associated data and secret key.

3.2 Keywords

can: A keyword indicating a capability (*can* equals *is able to*).

required: *See:* **shall**.

may: A keyword indicating a course of action permissible within the limits of this standard (*may* equals *is permitted to*).

must: A keyword used only to describe an unavoidable situation that does not constitute a requirement for compliance to this standard.

shall: A keyword indicating a mandatory requirement strictly to be followed in order to conform to this standard and from which no deviation is permitted (*shall* equals *is required to*).

shall not: A phrase indicating an absolute prohibition of this standard.

should: A keyword indicating that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended to*).

[1] NIST publications are available from the National Institute of Standards and Technology, NIST Public Inquiries, NIST, 100 Bureau Drive, Stop 3460, Gaithersburg, MD, 20899-3460, USA (www.nist.gov).
[2] *The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

3.3 Acronyms and abbreviations

AES	Advanced Encryption Standard (see NIST FIPS 197[3])
CTR	counter (as in AES counter mode of operation)
FIPS	Federal Information Processing Standard
GF	Galois Field (see Menezes et al. [B4][4])
LBA	logical block address
GCM	Galois/Counter Mode of authenticated encryption

4. Mathematical conventions

This standard uses decimal, binary, and hexadecimal numbers. For clarity, decimal numbers generally represent counts, and binary or hexadecimal numbers describe bit patterns or raw binary data.

Binary numbers are represented by a string of one or more binary digits, followed by the subscript 2. For example, the decimal number 26 is represented as 00011010_2 in binary.

Hexadecimal numbers are represented by a string of one or more hexadecimal digits, prefixed by the string "0x." For example, the decimal number 135 is represented as 0x87 in hexadecimal.

The number of bits in a bit string X is denoted as #X.

The procedures used in this standard are as follows:

- AES-Enc(*K, X*), the AES block cipher encryption (see NIST FIPS 197), returns the encryption of the input *X* with the key *K*.
- AES-Dec(*K, X*), the AES block cipher decryption, returns the decryption of the input *X* with the key *K*.
- mult-by-alpha(*X*), the multiplication-by-alpha of a 16-byte value $X \in$ GF (2^{128}) by a primitive element α in the field GF(2^{128}), defined in 5.2.1.
- $X \cdot Y$, the multiplication over the field GF (2^{128}) of two elements $X, Y \in$ GF (2^{128}), defined in 5.3.3.
- $X \oplus Y$, the addition of two elements $X, Y \in$ GF (2^{128}) that is equivalent to the bitwise exclusive-or operation.
- bitlength(*X*) returns an 8-byte string containing the non-negative integer describing the number of bits in the input *X*, with the most significant bit first and the least significant bit last. The expression 0^n denotes a string of *n* zero bits, and *A*|*B* denotes the concatenation of two bit strings *A* and *B*.
- bytelength(*X*) returns as an integer the number of bytes in the input X.
- $\text{msb}_t(X)$ returns the initial *t* bits of the input *X*. Bit strings are indexed starting from the most significant bit, so that bit zero of *X* is the most significant bit. *X*[*a*:*b*] denotes the substring of *X* from the *a*th bit through the *b*th.

[3] Information on references can be found in Clause 2.
[4] The numbers in brackets correspond to those of the bibliography in Annex A.

5. Wide-block encryption algorithms

5.1 Encryption with associated data

5.1.1 Data units and associated data

This standard specifies length-preserving encryption with associated data (EAD) algorithms that are suitable for the encryption of data at rest. An EAD algorithm consists of an encryption procedure and a decryption procedure. The encryption procedure accepts three inputs as follows:

— Secret key

— Plaintext

— Associated data

It returns a single ciphertext value. Each of these inputs is a byte string.

The security of these procedures depends on the secret key consisting of random or pseudo-random data that is unpredictable to the adversary. Each EAD algorithm accepts a key of a fixed length, but different algorithms may have keys of different lengths.

The plaintext input contains the data to be encrypted. Plaintext is divided into data units that, within a particular key scope, may have different lengths. An EAD algorithm defines the range of admissible plaintext lengths. A compliant implementation should keep the size of the data units the same within a key scope. If an implementation allows variable-length data units within a key scope, then the implementation's design should be such that it is able to unambiguously determine the length of each data unit without using data that is potentially under the control of an adversary. A compliant implementation shall use distinct associated data for distinct data units.

The associated data input contains data that is associated with the plaintext, but that does not need to be encrypted. The choice of data for this input is described in more detail as follows. Within a particular key scope, associated data units may have different lengths.

The ciphertext returned by the encryption procedure is the same length as the plaintext.

The decryption procedure accepts three inputs as follows:

— Secret key

— Ciphertext

— Associated data value

It returns a single plaintext value. The decryption procedure is the reverse of the encryption procedure; more specifically, if the encryption of the plaintext P with the key K and the associated data A results in the ciphertext C, then the decryption of C with the key K and the associated data A results in the plaintext P.

A conforming implementation shall include in the associated data only information that is available, in plaintext form, at the time of encryption and the time of decryption.

The associated data input shall uniquely identify the plaintext. This is because whenever the same plaintext is encrypted two different times using the same key but with distinct associated data values, the result is

two distinct ciphertext values. Thus the use of distinct associated data values hides the equality of the plaintexts from an attacker.

5.1.2 Using EAD to protect a string of data blocks

An EAD may be used to protect a string of data blocks, such as those in a data-storage disk. In this application, the associated data input to the encryption and decryption procedure should contain the logical index of the block on which the procedure is acting. When this information is included in the associated data, cases in which two distinct data blocks contain identical plaintext values are hidden from an adversary. Figure 1 shows an example of how an EAD performs encryption and decryption.

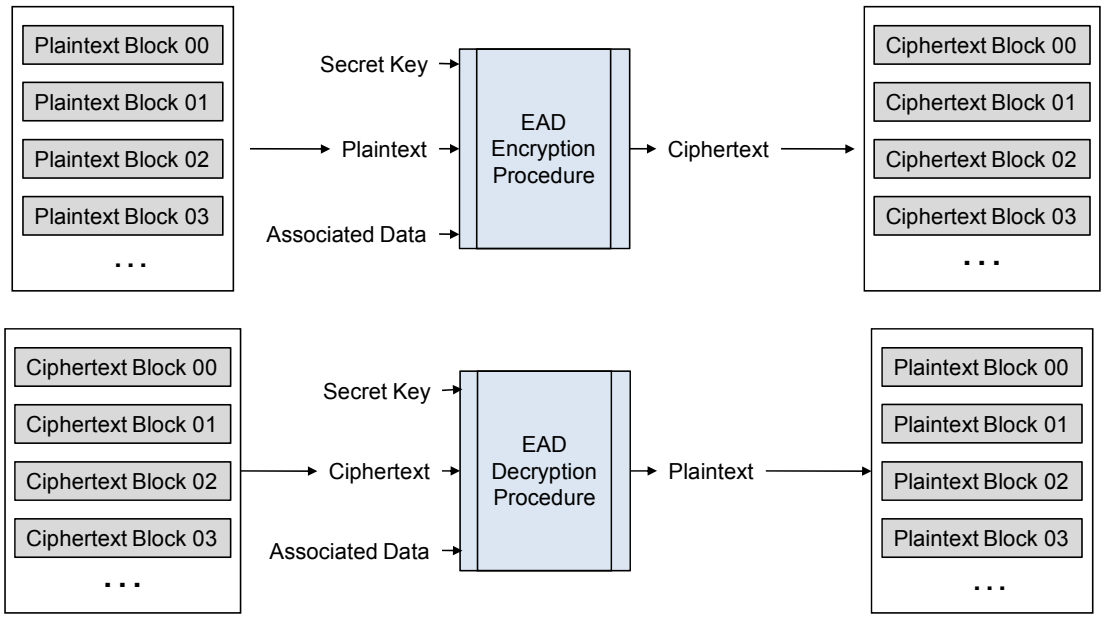

Figure 1—EAD encryption and decryption of a block device

If multiple disks are being protected with a single secret key, then the associated data input should contain both the logical index of the block and an additional distinguishing parameter that is unique to each of the disks. When this information is included in the associated data, cases in which two distinct data blocks on different disks contain identical plaintext values are hidden from an adversary.

5.2 EME2-AES algorithm

5.2.1 The mult-by-alpha procedure

The encryption and decryption procedures described in the following subclauses use a mult-by-alpha(X) procedure that multiplies a 16-byte value X by a primitive element α in the field GF(2^{128}). The input value is first converted into a byte string $X[i]$, $i = 0, 1, \ldots, 15$, where $X[0]$ is the first byte of the byte string.

The multiplication by alpha is defined in Table 1 or by a mathematically equivalent procedure.

Table 1—The mult-by-alpha(X) procedure

```
// compute mult-by-alpha(X), where X is a 16-byte value
//
// X[i] denotes the ith byte of X, and indexing starts at 0
//
// i is an integer used in loops
//
// Y[i] denotes the ith byte of the output Y, and indexing starts at 0

1.  for i = 0 to 15 do
2.     Y[i] = 2*X[i] mod 256
3.     if (i>0 and X[i-1]>127) then Y[i]=Y[i]+1
4.  end-for
5.  if (X[15] > 127) then Y[0] = Y[0] ⊕ 0x87
6.  return Y
```

Conceptually, the procedure is a left shift of each byte by one bit with carry propagating from one byte to the next. Also, if the 15th (last) byte shift results in a carry, a special value (hexadecimal 0x87) is xor'ed into the first byte. This value is derived from the modulus of the Galois Field (polynomial $x^{128}+x^7+x^2+x+1$).

5.2.2 EME2-AES encryption

Equation (1) describes the EME2-AES encryption procedure:

$$C = \text{EME2-AES-Enc}(Key, T, P) \tag{1}$$

where

Key is the 48- or 64-byte EME2-AES key
T is the associated data, of arbitrary byte length (zero or more bytes)
P is the plaintext, of length 16 bytes or more
C is the ciphertext resulting from the operation, of the same byte-length as *P*

The input to the EME2-AES encryption procedure is parsed as follows:

— The key is partitioned into three fields, $Key = K_{AD} \mid K_{ECB} \mid K_{AES}$, with K_{AD} (the associated data key) consisting of the first 16 bytes, K_{ECB} (the ECB pass key) consisting of the following 16 bytes, and K_{AES} (the AES encryption/decryption key) consisting of the remaining 16 or 32 bytes.

— If not empty, the associated data T is partitioned into a sequence of blocks $T = T_1 \mid T_2 \mid \ldots \mid T_r$, where each of the blocks $T_1, T_2, \ldots, T_{r-1}$ is of length exactly 16 bytes, and T_r is of length between 1 and 16 bytes.

— The plaintext P is partitioned into a sequence of blocks $P = P_1 \mid P_2 \mid \ldots \mid P_m$, where each of the blocks $P_1, P_2, \ldots, P_{m-1}$ is of length exactly 16 bytes, and P_m is of length between 1 and 16 bytes.

As part of the ciphertext computation, the associated data T are processed by the sequence of steps in Table 2 or by a mathematically equivalent procedure.

Table 2—The function H for processing the associated data *T*

```
// Function H(K_AES, K_AD, T = (T_1 | ... | T_r-1 | T_r) ) :
1.  if bytelength(T) == 0 then
2.    T_star = AES-Enc(K_AES, K_AD)
3.  else
4.    K_AD = mult-by-alpha(K_AD)
5.    for j = 1 to r-1
6.      TT_j = AES-Enc(K_AES, K_AD ⊕ T_j) ⊕ K_AD
7.      K_AD = mult-by-alpha(K_AD)
8.    next
9.    if bytelength(T_r) < 16 then
10.     T_r = T_r | 0x80 | 0x00 | ...   // pad T_r to 16 bytes, 0x80 followed by 0's
11.     K_AD = mult-by-alpha(K_AD)
12.   endif
13.   TT_r = AES-Enc(K_AES, K_AD ⊕ T_r) ⊕ K_AD
14.   T_star = TT_1 ⊕ TT_2 ⊕ … ⊕ TT_r
15. endif

16. return T_star                    // return the 16 byte value T_star
```

A conforming implementation shall compute the ciphertext by executing the sequence of steps in Table 3 or a mathematically equivalent procedure.

Table 3—The EME2-AES encryption procedure

```
// EME2-AES-Enc(K_AES, K_ECB, K_AD, T, P = (P_1 | ... | P_m-1 | P_m) ):
1.  T_star = H(K_AES, K_AD, T)       // Process the associated data
2.  if bytelength(P_m) == 16 then
3.    lastFull = m
4.  else
5.    lastFull = m-1
6.    PPP_m = P_m | 0x80 | 0x00 | ...  // Pad P_m to 16 bytes, 0x80 followed by 0's
7.  endif

//     First ECB pass
8.  L = K_ECB
9.  for j = 1 to lastFull
10.   PPP_j = AES-Enc(K_AES, L ⊕ P_j)
11.   L = mult-by-alpha(L)
12. next

//     Intermediate mixing
13. MP = PPP_1 ⊕ PPP_2 ⊕ ... ⊕ PPP_m ⊕ T_star
14. if bytelength(P_m) < 16 then
15.   MM = AES-Enc(K_AES, MP)
16.   MC = MC_1 = AES-Enc(K_AES, MM)
17. else
18.   MC = MC_1 = AES-Enc(K_AES, MP)
19. endif
20. M = M_1 = MP ⊕ MC
21. for j = 2 to lastFull
22.   if (j-1 mod 128 > 0) then      // use the current mask M
23.     M = mult-by-alpha(M)
24.     CCC_j = PPP_j ⊕ M
25.   else                          // calculate a new mask M
26.     MP = PPP_i ⊕ M_1
27.     MC = AES-Enc(K_AES, MP)
28.     M = MP ⊕ MC
29.     CCC_j = MC ⊕ M_1
30.   endif
31. next
32. if lastFull < m then
33.   C_m = P_m ⊕ (MM truncated to bytelength(P_m) bytes)
34.   CCC_m = C_m | 0x80 | 0x00 | ...  // Pad C_m to 16 bytes, 0x80 followed by 0's
35. endif
36. CCC_1 = MC_1 ⊕ CCC_2 ⊕ ... ⊕ CCC_m ⊕ T_star


//     Second ECB Pass
37. L = K_ECB
38. for j = 1 to lastFull
39.   C_j = AES-Enc(K_AES, CCC_j) ⊕ L
40.   L = mult-by-alpha(L)
41. next

42. return C = (C_1 | ... | C_m-1 | C_m)
```

An illustration of the EME2-AES encryption procedure (for plaintext of 130 full blocks and one partial block) is provided in Figure 2.

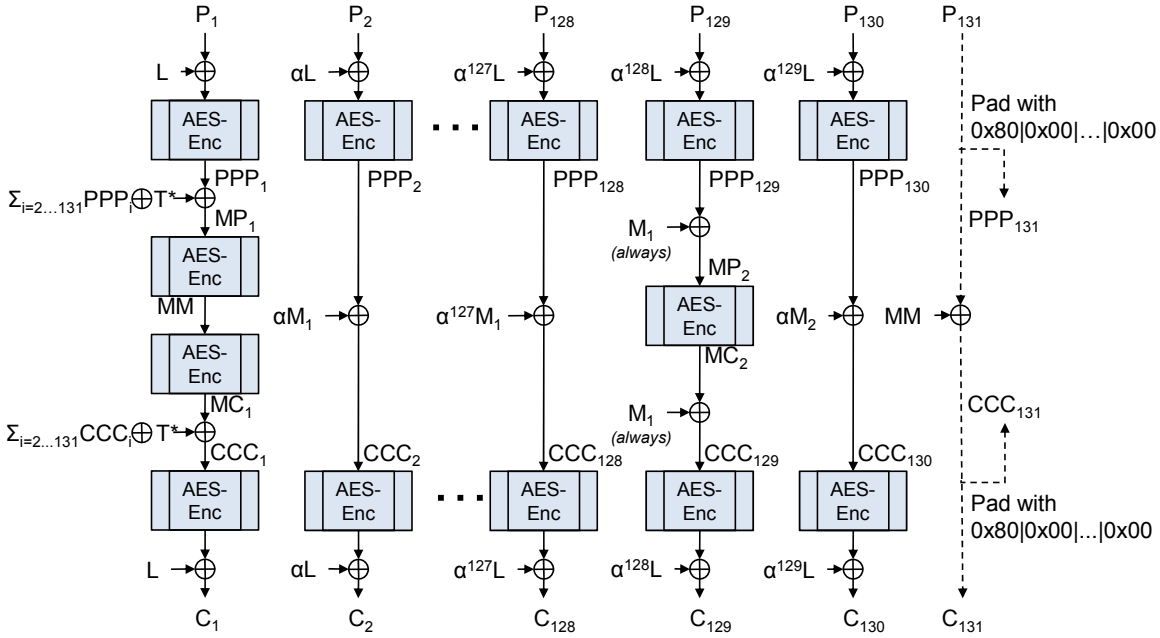

Figure 2—An illustration of EME2-AES encryption

In Figure 2, each AES-Enc block has a Key input that is not shown. *T** is computed from the associated data using the function H from Table 2, and the M_i's are computed as $M_i = MP_i$ xor MC_i.

5.2.3 EME2-AES decryption

Equation (2) describes the EME2-AES decryption procedure:

$$P = \text{EME2-AES-Dec}(Key, T, C) \tag{2}$$

where

Key is the 48- or 64-byte EME2-AES key
T is the associated data, of arbitrary byte length (zero or more bytes)
C is the ciphertext, of length 16 bytes or more
P is the plaintext resulting from the operation, of the same byte-length as *C*

The input to the EME2-AES decryption procedure is parsed as follows:

— The key is partitioned into three fields, $Key = K_{AD} \mid K_{ECB} \mid K_{AES}$, with K_{AD} (the associated data key) consisting of the first 16 bytes, K_{ECB} (the ECB pass key) consisting of the following 16 bytes, and K_{AES} (the AES encryption/decryption key) consisting of the remaining 16 or 32 bytes.

— If not empty, the associated data is partitioned into a sequence of blocks $T = T_1 \mid T_2 \mid \ldots \mid T_r$, where each of the blocks $T_1, T_2, \ldots, T_{r-1}$ is of length exactly 16 bytes, and T_r is of length between 1 and 16 bytes.

— The ciphertext C is partitioned into a sequence of blocks $C = C_1 \mid C_2 \mid \ldots \mid C_m$, where each of the blocks $C_1, C_2, \ldots, C_{m-1}$ is of length exactly 16 bytes, and C_m is of length between 1 and 16 bytes.

A conforming implementation shall compute the plaintext by executing the sequence of steps in Table 4 or a mathematically equivalent procedure.

NOTE—The only difference between the encryption and decryption procedures is that all the AES-Enc calls in Table 3 are replaced in Table 4 by calls to AES-Dec. The function H is defined in Table 2.[5]

[5] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

Table 4—The EME2-AES decryption procedure

```
// EME2-AES-Dec(K_AES, K_ECB, K_AD, T, C = (C_1 | ... | C_m-1 | C_m) ):
1.  T_star = H(K_AES, K_AD, T)       // Process the associated data
2.  if bytelength(C_m) == 16 then
3.     lastFull = m
4.  else
5.     lastFull = m-1
6.     CCC_m = C_m | 0x80 | 0x00 | ...  // Pad C_m to 16 bytes, 0x80 followed by 0's
7.  endif

//  First ECB pass
8.  L = K_ECB
9.  for j = 1 to lastFull
10.    CCC_j = AES-Enc(K_AES, L ⊕ C_j)
11.    L = mult-by-alpha(L)
12. next

//  Intermediate mixing
13. MC = CCC_1 ⊕ CCC_2 ⊕ ... ⊕ CCC_m ⊕ T_star
14. if bytelength(C_m) < 16 then
15.    MM = AES-Dec(K_AES, MC)
16.    MP = MP_1 = AES-Dec(K_AES, MM)
17. else
18.    MP = MP_1 = AES-Dec(K_AES, MC)
19. endif
20. M = M_1 = MP ⊕ MC
21. for j = 2 to lastFull
22.    if (j-1 mod 128 > 0) then     // use the current mask M
23.       M = mult-by-alpha(M)
24.       PPP_j = CCC_j ⊕ M
25.    else                          // calculate a new mask M
26.       MC = CCC_j ⊕ M_1
27.       MP = AES-Dec(K_AES, MC)
28.       M = MP ⊕ MC
29.       PPP_j = MP ⊕ M_1
30.   endif
31. next
32. if lastFull < m then
33.    P_m = C_m ⊕ (MM truncated to bytelength(C_m) bytes)
34.    PPP_m = P_m | 0x80 | 0x00 | ...  // Pad P_m to 16 bytes, 0x80 followed by 0's
35. endif
36. PPP_1 = MP_1 ⊕ PPP_2 ⊕ ... ⊕ PPP_m ⊕ T_star

//  Second ECB Pass
37. L = K_ECB
38. for j = 1 to lastFull
39.    P_j = AES-Dec(K_AES, PPP_j) ⊕ L
40.    L = mult-by-alpha(L)
41. next

42. return P = (P_1 ... P_m-1 P_m)
```

5.3 XCB-AES algorithm

5.3.1 Overview of XCB-AES

The XCB-AES algorithm use the AES block cipher encryption procedures AES-Enc and AES-Dec, as well as the hash function h and the pseudorandom function c. The variables H, K_e, K_d, and K_c are derived from K, essentially by running the AES-Enc encryption procedure in CTR mode (see NIST SP 800-38A). A conforming implementation may store these values between computation of encryption and decryption procedures, in order to trade off storage and computational load.

5.3.2 The function c

Let k be the size of the key fed to the AES procedure (either 16 or 32 bytes).

The function **c**: $\{0, 255\}^k \times \{0, 255\}^{16} \rightarrow \{0, 255\}^l$, where the output length l (in bytes) is bounded by $0 <= l <= 2^{36}$, generates an arbitrary-length output by running the AES-Enc procedure in CTR mode, using its 16-byte input as the initial counter value. Its definition is as follows:

$$\mathbf{c}(K, W, l) = \text{AES-Enc}(K, W) \mid \text{AES-Enc}(K, \text{incr}(W)) \mid \ldots \mid \text{msb}_t(\text{AES-Enc}(K, \text{incr}^{n-1}(W)))$$

where the output length l is indicated as an explicit parameter for clarity; $n = \lceil l / 16 \rceil$ is the number of 16-byte blocks in the output, and $t = l \bmod 16$ is the number of bytes in the trailing block.

Here the function incr: $\{0, 255\}^{16} \rightarrow \{0, 255\}^{16}$ is the increment operation that is used to generate successive counter values. This function treats the rightmost 4 bytes of its argument as a non-negative integer with the least significant byte on the right, and increments this value modulo 2^{32}. More formally,

$$\text{incr}(X) = X[0:95] \mid (X[96:127] + 1 \bmod 2^{32})$$

where bit strings are implicitly converted into integers.

5.3.3 Multiplication in GF(2^{128})

The multiplication operation of two 16-byte values X, $Y \in \text{GF}(2^{128})$ is mathematically equivalent to an operation on bit vectors. The result $Z = X \cdot Y$ is also an element of GF(2^{128}). The input values are first converted into a byte string $X[i]$, $i = 0,1,...,15$, where the most significant byte is $X[0]$, and the least significant byte is $X[15]$.

The multiplication in GF(2^{128}) operation is defined in Table 5 or by a mathematically equivalent procedure:

Table 5—The multiplication in GF(2^{128}) procedure

```
// mult-GF(X, Y)

// compute Z = X * Y, where X, Y, Z are elements of GF(2^128) using a
// byte based algorithm
//
// here A[i] denotes the ith byte of A, and indexing starts at 0
//
// i and j are integers used in loops
// mask, b, and msb are unsigned integers

   /* initialize z to the all-zero element */
7.  for i = 0 to 15
8.     z[i] = 0
9.  next

10. for i=0 to 15   /* loop over bytes of y */
11.    mask = 128
12.    while mask > 0       /* loop over bits in byte */
              /* if masked bit is set, add in terms from x */
13.       b = y[i] & mask
14.       if b != 0
15.          for j=0 to 15
16.             set z[j] to z[j] ^ x[j]
17.          next
18.       endif
19.       mask = mask / 2;
              /* now execute LFSR shift on x  */
20.       set msb to x[15] & 0x01
21.       for j=15 down to 1
22.          set b to x[j-1] & 0x01
23.          set x[j] to (x[j] / 2) + b * 128
24.       next
25.       set x[0] to x[0] / 2
26.       if msb == 1
27.          set x[0] to x[0] ^ 0xe1
28.       endif
29.    endwhile
30. next
31. return z
```

NOTE—In Table 5, the multiplication operation uses the special element $R = 11100001_2|0^{120}$ (0xe1 | 0x00 | … | 0x00).

5.3.4 The procedures h_1 and h_2

The procedures h_1 and h_2 are defined in terms of the underlying hash function h as follows:

$$h_1 (H, Z, B) = h(H, 0^{128} \mid Z, B \mid 0^{padlen1(\#B)})$$

$$h_2 (H, Z, B) = h(H, Z \mid 0^{128}, B \mid 0^{padlen2(\#B)} \mid bitlength(Z \mid 0^{128}) \mid bitlength(B))$$

The procedure padlen2(x) returns the integer between 0 and 127, inclusive, that when added to x is a multiple of 128. The procedure padlen1(x) returns padlen2(x) + 128. Equation (3) and Equation (4) express these statements mathematically:

$$\text{padlen2}(x) = 128 * \text{ceil}(x/128) - x \tag{3}$$

$$\text{padlen1}(x) = 128 * (1 + \text{ceil}(x/128)) - x = 128 + \text{padlen2}(x) \tag{4}$$

where ceil(x) is the smallest integer that is larger than or equal to x.

The procedure $h : \{0, 1\}^{128} \times \{0, 1\}^{a} \times \{0, 1\}^{c} \rightarrow \{0, 1\}^{128}$ is defined by $h(H, A, C) = X_{m+n+1}$, where a and c are within the interval $[128, 2^{39}]$, m is the number of 16-byte blocks in A, n is the number of 16-byte blocks in C, and the variables $X_i \in \{0, 1\}^{128}$ for $i = 0, \ldots, m+n+1$ are defined as follows:

$$X_i = 0 \qquad \text{for } i = 0$$

$$X_i = (X_{i-1} \oplus A_i) \cdot H \qquad \text{for } i = 1, \ldots, m-1$$

$$X_i = (X_{m-1} \oplus A_m) \cdot H \qquad \text{for } i = m$$

$$X_i = (X_{i-1} \oplus C_{i-m}) \cdot H \qquad \text{for } i = m+1, \ldots, m+n-1$$

$$X_i = (X_{m+n-1} \oplus C_n) \cdot H \qquad \text{for } i = m+n$$

$$X_i = (X_{m+n} \oplus (\text{bitlength}(A) \mid \text{bitlength}(C))) \cdot H \qquad \text{for } i = m+n+1$$

Let A_i denote the 16-byte substring $A[128*(i-1): 128*i-1]$, and let C_i denote $C\ [128*(i-1): 128*i-1]$. In other words, A_i and C_i are the ith blocks of A and C, respectively, if those bit strings are decomposed into 16-byte blocks (the last A_i and C_i blocks are padded with 0s, if shorter then 16 bytes).

NOTE—This procedure is identical to the universal hash function that is used as a component of the Galois/Counter Mode (GCM) of Operation (see NIST SP 800-38D [B5]). It is equivalent to the procedure used in Step 5 of Algorithm 4 of that specification, but please note that it is different than GHASH as defined in that document.

5.3.5 XCB-AES encryption

The XCB-AES encryption procedure for an m-bit block P is modeled with Equation (5):

$$CT \leftarrow \text{XCB-AES-Enc}(K, P, Z) \tag{5}$$

where

K is the 16- or 32-byte XCB-AES key
P is the plaintext of m bits where $m \in [128, 2^{32}]$, and m is a multiple of 8
Z is the value of the associated data, of arbitrary byte length (zero or more bytes)
CT is the ciphertext resulting from the operation, of the same length as P

A conforming implementation shall compute the ciphertext by executing the following steps or a mathematically equivalent procedure (see Figure 3):

$H \leftarrow \text{AES-Enc}(K, 0^{128})$

$K_e \leftarrow \text{msb}_k(\text{AES-Enc}(K, 0^{125}|001_2)| \text{ AES-Enc}(K, 0^{125}|010_2))$

$K_d \leftarrow \text{msb}_k(\text{AES-Enc}(K, 0^{125}|011_2)| \text{ AES-Enc}(K, 0^{125}|100_2))$

$K_c \leftarrow \text{msb}_k(\text{AES-Enc}(K, 0^{125}|101_2)| \text{ AES-Enc}(K, 0^{125}|110_2))$

$A \leftarrow P[m\text{–}128: m\text{–}1]$

$B \leftarrow P[0: m\text{–}127]$

$C \leftarrow \text{AES-Enc}(K_e, A)$

$D \leftarrow C \oplus \text{h}_1(H, Z, B)$

$E \leftarrow B \oplus \text{c}(K_c, D, \#B)$

$F \leftarrow D \oplus \text{h}_2(H, Z, E)$

$G \leftarrow \text{AES-Dec}(K_d, F)$

$CT \leftarrow E|G$

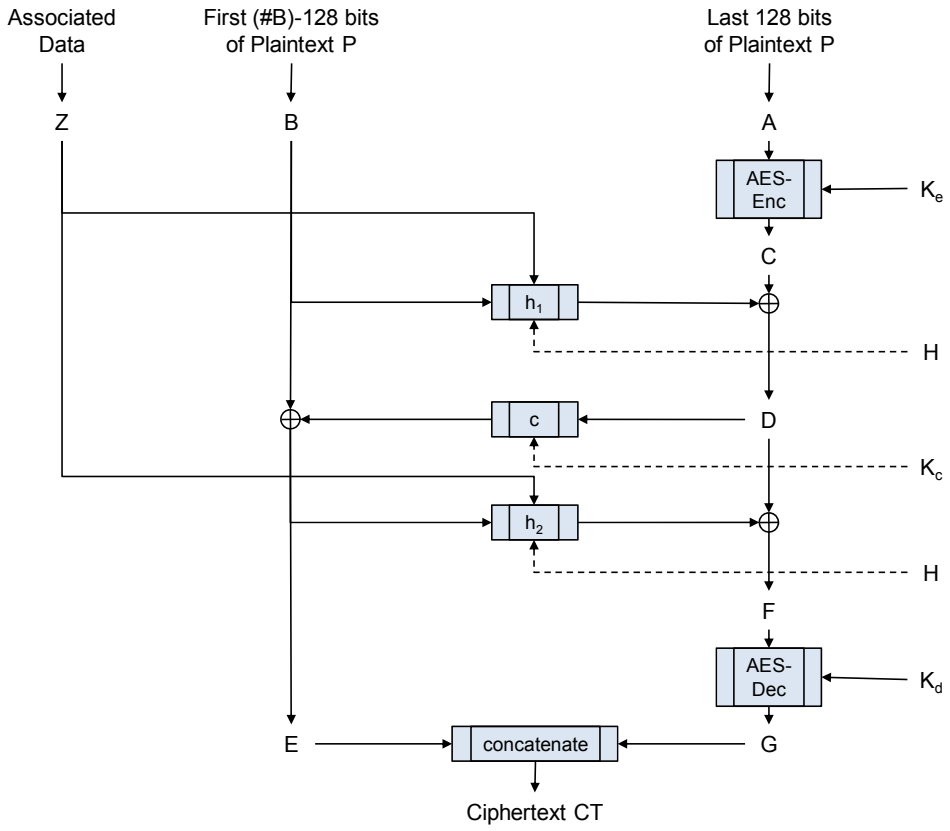

Figure 3—An illustration of XCB-AES encryption

5.3.6 XCB-AES decryption

The XCB-AES decryption procedure for an *m*-bit block *CT* is modeled with Equation (6):

$$P \leftarrow \text{XCB-AES-Dec}(K, CT, Z) \tag{6}$$

where

K is the 16- or 32-byte XCB-AES key
CT is the ciphertext of *m* bits where $m \in [128, 2^{32}]$, and *m* is a multiple of 8
Z is the associated data, of arbitrary byte length (zero or more bytes)
P is the plaintext resulting from the operation, of the same length as *CT*

A conforming implementation shall compute the plaintext by executing the following sequence of steps or a mathematically equivalent procedure (see Figure 4):

$H \leftarrow \text{AES-Enc}(K, 0^{128})$

$K_e \leftarrow \text{msb}_k(\text{AES-Enc}(K, 0^{125}|001_2)| \text{ AES-Enc}(K, 0^{125}|010_2))$

$K_d \leftarrow \text{msb}_k(\text{AES-Enc}(K, 0^{125}|011_2)| \text{ AES-Enc}(K, 0^{125}|100_2))$

$K_c \leftarrow \text{msb}_k(\text{AES-Enc}(K, 0^{125}|101_2)| \text{ AES-Enc}(K, 0^{125}|110_2))$

$G \leftarrow CT[m–128: m–1]$

$E \leftarrow CT[0: m–127]$

$F \leftarrow \text{AES-Enc}(K_d, G)$

$D \leftarrow F \oplus h_2(H, Z, E)$

$B \leftarrow E \oplus c(K_c, D, \#E)$

$C \leftarrow D \oplus h_1(H, Z, B)$

$A \leftarrow \text{AES-Dec}(K_e, C)$

$P \leftarrow B|A$

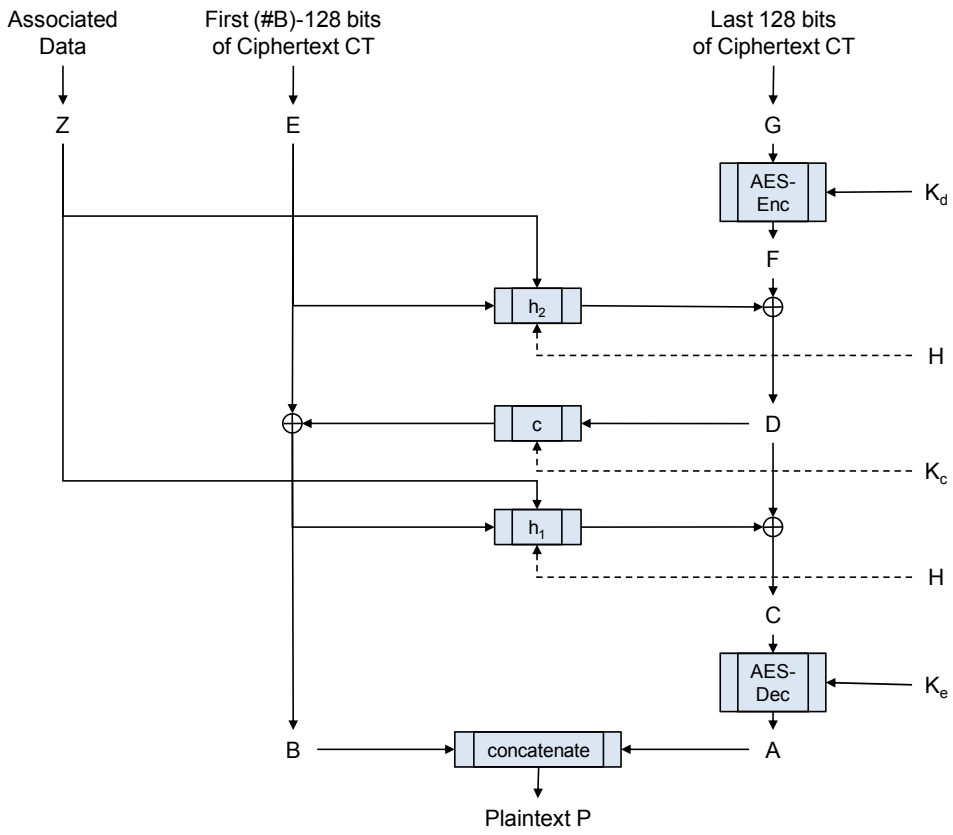

Figure 4—An illustration of XCB-AES decryption

6. Compliance

The EME2-AES encryption and decryption procedures described in 5.2.2 and 5.2.3 use AES as the basic building block with a key of either 48 or 64 bytes. EME2-AES-128 uses a 48-byte key and EME2-AES-256 uses a 64-byte key.

The XCB-AES encryption and decryption procedures described in 5.3.3 and 5.3.6 use AES as the basic building block with a key of either 16 or 32 bytes. XCB-AES-128 uses a 16-byte key and XCB-AES-256 uses a 32-byte key.

An implementation may claim compliance with this standard if it fulfills all the requirements for either the encryption or decryption procedures or both of at least one of the following algorithms:

— EME2-AES-128

— EME2-AES-256

— XCB-AES-128

— XCB-AES-256

Annex A

(informative)

Bibliography

[B1] Halevi, S., "EME*: extending EME to handle arbitrary-length messages with associated data," *INDOCRYPT 2004,* Lecture Notes in Computer Science, vol. 3348, pp. 315–327. Springer-Verlag, 2004.

[B2] IEEE Std 1619™-2007, IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices.[6, 7]

[B3] McGrew, D., and Fluhrer, S., "The Security of the Extended Codebook (XCB) Mode of Operation," *Proceedings of the 14th Annual Workshop on Selected Areas in Cryptography,* Springer-Verlag, 2007.

[B4] Menezes, A. J., Van Oorschot, P. C., Vanstone, S. A., *Handbook of Applied Cryptography,* CRC Press, October 1996.

[B5] NIST Special Publication 800-38D (NIST SP 800-38D), Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.

[6] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://shop.ieee.org/).

[7] The IEEE standards or products referred to in Annex A are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

Annex B

(informative)

Implementation guidance

B.1 Security considerations

The security goal of length-preserving EAD is roughly described as follows: first, fixing the plaintext length and any particular value for the associated data, the EAD scheme should look just like a block cipher with block size equal to the plaintext size. Namely, interacting with the encryption and decryption routines, it should be infeasible to distinguish them from a random permutation and its inverse. Moreover, varying the plaintext length and/or the associated data should look like using a different and independent key.

A little more precisely, the security model stipulates an attacker that is capable to request the encryption of plaintext/associated-data pairs under an unknown key, and is similarly capable to request the decryption of ciphertext/associated-data pairs under the same unknown key. The adversary is capable to adaptively choose the plaintext, ciphertext, and associated-data values. The security requirement asserts that this attacker is not able to distinguish between the following two cases:

a) The queries are indeed answered by the scheme at hand with a fixed secret key.

b) The queries are answered by a set of random permutations, with different independent permutations for different values of the associated data and plaintext length.

This security requirement implies, in particular, that encrypting some plaintext with one value of the associated data and then decrypting the resulting ciphertext using a different value of the associated data yields a (pseudo)random decrypted plaintext [since this is what happens in Case b)].

Both EME2-AES and XCB-AES have been shown to be secure in this model, under the assumption that AES block cipher encryption (see NIST FIPS 197) is not distinguishable from a random permutation (and subject to some bound on the number of invocations with the same key—roughly the birthday bound).

This security model assumes that the encryption/decryption routines are never applied to plaintext/ciphertext values that relate directly to the secret keys. In particular this model does not consider the case where the encryption routine of the scheme is used to encrypt its own secret key.

B.2 Performance characteristics of EME2-AES and XCB-AES

This document specifies two different EAD algorithms: EME2-AES and XCB-AES. Both implement a tweakable pseudorandom permutation with substantially similar security properties and have similar bounds with respect to the amount of data that is able to be safely be encrypted with a single key.

Nevertheless, upon choosing an algorithm, implementers might need to consider other factors than security level such as software performance or hardware implementation size.

Table B.1 shows a list of potentially significant differences, in term of computational complexity, between the two algorithms, when encrypting data blocks made of n 16-byte blocks:

Table B.1—Differences, in term of computational complexity, between EME2-AES and XCB-AES

	EME2-AES-128	EME2-AES-256	XCB-AES-128	XCB-AES-256
Applications of the AES procedure	2*n*+1	2*n*+1	*n*+1	*n*+1
Shift and xor operations (multiplication by a fixed alpha)	3*n*	3*n*	0	0
GF(2^{128}) multiplications	0	0	2*n*	2*n*
Key storage (bytes)	48	64	64	112

B.3 Application to logical block-level disk encryption

In an application of this standard to logical block-level encryption of a disk:

a) The data unit typically corresponds to a logical block,

b) The key scope typically includes a range of consecutive logical blocks on the disk, and

c) The associated data value corresponding to the first data unit in the scope typically corresponds to the Logical Block Address (LBA) associated with the logical block in the range.

The associated data values are assigned consecutively, starting from an arbitrary non-negative integer. When encrypting an associated data value using AES, the associated data value is first converted into a little-endian byte string. For example the associated data value 0x123456789a corresponds to byte string 0x9a, 0x78, 0x56, 0x34, 0x12.

A conforming implementation shall not use a key for wide-block encryption of storage with more than one key scope.

NOTE—The reason for the preceding restriction is that encrypting more than one block with the same key and the same associated data value introduces security vulnerabilities that might potentially be used in an attack on the system. In particular, key reuse enables trivial cut-and-paste attacks.

Annex C

(informative)

Test vectors

C.1 Encoding

Test vectors are encoded in C-array format in order to facilitate their use in C-code implementations. Each test vector is represented between brackets, and the following number is the number of bytes. The C struct in use is shown in Table C.1.

Table C.1—C struct used to represent test vectors

```
typedef struct test_case_t {
  uint8_t key[64];    /* holds AES keys of any size (16, 24, 32, 64) */
  uint8_t bytes_in_key;
  uint8_t plaintext[TEST_BUF_LEN];
  unsigned int plaintext_len;
  uint8_t assoc_data[TEST_BUF_LEN];
  unsigned int assoc_data_len;
  uint8_t ciphertext[TEST_BUF_LEN];
  struct test_case_t *next;
} test_case_t;
```

C.2 EME2-AES test vectors

Table C.2 shows 10 test vectors referred to as “case1” through “case 9” testing EME2-AES with different combinations of plaintext, key length, and associated data lengths.

Table C.2—EME2-AES Test Cases

```
test_case_t case1 = {
  /* key */
  {
    0xd6, 0xc2, 0xb8, 0x55, 0x32, 0xb0, 0x2e, 0xec,
    0x5a, 0x0e, 0x7e, 0xfc, 0x1a, 0x31, 0xdb, 0xef,
    0x2f, 0x4e, 0xcf, 0x01, 0xeb, 0x2a, 0xfd, 0x5d,
    0xed, 0x4e, 0x71, 0x77, 0xc0, 0xd4, 0x1e, 0x07,
    0x45, 0x20, 0x78, 0x6f, 0x13, 0xef, 0x99, 0xb1,
    0xe6, 0x2a, 0x98, 0xd2, 0x5b, 0x6e, 0xc5, 0x16,
    0x8d, 0xcf, 0xe3, 0xb4, 0x53, 0xb9, 0x68, 0xc5,
    0x4f, 0x73, 0x62, 0xf2, 0xa4, 0x89, 0x19, 0x23,
  },
  /* bytes in key */
  64,
  /* plaintext */
  {
    0x18, 0x7d, 0xdd, 0x99, 0x8c, 0x75, 0xb5, 0x50,
    0xf8, 0x32, 0x94, 0x62, 0x77, 0x2a, 0xc0, 0xd6,
  },
  /* bytes in plaintext */
```

```
  16,
  /* associated data */
  {
  },
  /* bytes in associated data */
  0,
  /* ciphertext */
  {
    0x60, 0x99, 0x53, 0xbc, 0x9a, 0x86, 0xf7, 0x1a,
    0xee, 0x0c, 0xed, 0x2e, 0xee, 0x7f, 0x03, 0xdf,
  },
  &case4
};

test_case_t case2 = {
  /* key */
  {
    0x36, 0x7b, 0x9c, 0x49, 0xdd, 0xf2, 0xfa, 0xde,
    0x0b, 0x02, 0x5e, 0x1c, 0x95, 0xbe, 0xb3, 0x7a,
    0x5b, 0x1e, 0x6c, 0x42, 0x8d, 0x45, 0xf1, 0x9d,
    0xde, 0xe0, 0x63, 0xd2, 0xf5, 0xe9, 0x56, 0x33,
    0x9d, 0x15, 0x0e, 0x07, 0x7a, 0xfd, 0xe4, 0xb8,
    0x29, 0x53, 0x32, 0x1c, 0x7f, 0x18, 0x0c, 0x2b,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x53, 0xa3, 0x7f, 0xc1, 0x96, 0xa6, 0xc4, 0xaa,
    0xfc, 0x05, 0xee, 0xc2, 0xad, 0xe7, 0x50, 0x1c,
  },
  /* bytes in plaintext */
  16,
  /* associated data */
  {
    0xfb, 0x37, 0xa0, 0x28, 0xd3, 0x4a, 0x2e, 0x2f,
    0xc2, 0x11, 0xb7, 0xad, 0x62, 0x88, 0x6a, 0x6a,
    0x78, 0xb8, 0x7a, 0xf1,
  },
  /* bytes in associated data */
  20,
  /* ciphertext */
  {
    0x39, 0xbb, 0xae, 0xab, 0x50, 0xc4, 0xa0, 0x62,
    0xff, 0x62, 0xe2, 0xc6, 0xa2, 0xbe, 0x52, 0x64,
  },
  &case1
};

test_case_t case3 = {
  /* key */
  {
    0x41, 0x37, 0x1f, 0x79, 0xa2, 0x2b, 0x3c, 0xda,
    0x7c, 0x25, 0xb1, 0x4c, 0xe0, 0x3a, 0xdd, 0x54,
    0x23, 0x4f, 0xd8, 0xb2, 0xe1, 0xba, 0x9c, 0xc5,
    0x74, 0xc4, 0xc0, 0x46, 0xd7, 0x7a, 0x05, 0x9f,
    0x0a, 0x57, 0x97, 0xe4, 0x18, 0x21, 0xb6, 0x48,
```

```
    0x53, 0x23, 0x16, 0x9f, 0xbf, 0x42, 0xe1, 0x27,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x22, 0x1f, 0xd8, 0x2f, 0x37, 0x81, 0x64, 0x0b,
    0xb3, 0xf4, 0x35, 0x76, 0xc8, 0xe3, 0xf8, 0x33,
  },
  /* bytes in plaintext */
  16,
  /* associated data */
  {
    0xed, 0x7b, 0x85, 0x27, 0x2d, 0xbc, 0x00, 0x1f,
    0x96, 0x6d, 0xbb, 0xa8, 0x9a, 0x0e, 0xf1, 0x6a,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0xf2, 0x37, 0x8f, 0xe1, 0x98, 0x27, 0xc0, 0xb5,
    0xb3, 0xbc, 0xaf, 0xa5, 0x49, 0x11, 0xcb, 0x6b,
  },
  &case2
};

test_case_t case4 = {
  /* key */
  {
    0xfb, 0x7f, 0xfa, 0xa9, 0x0d, 0xa7, 0xfa, 0xc8,
    0x8d, 0xc9, 0xc0, 0x7e, 0xdf, 0x92, 0x59, 0xd0,
    0xc7, 0xcb, 0xa7, 0x6d, 0x50, 0x54, 0x18, 0xe3,
    0xee, 0x2b, 0xf5, 0xa8, 0xd4, 0x16, 0x10, 0xf7,
    0xd4, 0xf8, 0xe8, 0x3f, 0xd1, 0x7b, 0x95, 0x3b,
    0x21, 0x98, 0x46, 0xba, 0x88, 0x89, 0x2c, 0x68,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0xe7, 0x5e, 0x49, 0x46, 0x20, 0x64, 0x1a, 0x53,
    0xe0, 0xaa, 0xbd, 0xc9, 0x3c, 0x72, 0x60, 0x36,
    0xc8,
  },
  /* bytes in plaintext */
  17,
  /* associated data */
  {
    0x1e, 0x80, 0xfe, 0xca, 0x90, 0x9d, 0x75, 0x33,
    0x0e, 0xb5, 0x76, 0x01, 0xf8, 0x7c, 0x1d, 0xac,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0xaa, 0x01, 0xc6, 0x9d, 0x81, 0xd0, 0x61, 0xa6,
    0xf9, 0xa0, 0xda, 0x07, 0x1f, 0xc0, 0x9b, 0xa2,
```

```
    0x96,
  },
  &case3
};

test_case_t case5 = {
  /* key */
  {
    0xaa, 0xc0, 0xaa, 0xb3, 0x5d, 0x4b, 0x75, 0x62,
    0x6b, 0xef, 0x4e, 0x78, 0xde, 0x0c, 0xb1, 0x3e,
    0xbc, 0x1d, 0x03, 0x83, 0x68, 0x35, 0x69, 0xad,
    0x81, 0x9b, 0x79, 0xf8, 0xa4, 0x7f, 0xfe, 0xe6,
    0xe8, 0xdc, 0x58, 0x66, 0x0e, 0xb1, 0xb1, 0x7e,
    0xbc, 0x0f, 0xd9, 0x03, 0xed, 0x3c, 0x05, 0x2f,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x40, 0xc9, 0xec, 0xe1, 0xc6, 0x90, 0x2a, 0x89,
    0x0a, 0xd8, 0x78, 0xb2, 0x01, 0x46, 0x02, 0x3e,
    0x46, 0x5d, 0x2b, 0xe9,
  },
  /* bytes in plaintext */
  20,
  /* associated data */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0xe5, 0x7c, 0x72, 0x8d, 0xa8, 0x00, 0xc3, 0x64,
    0xf8, 0xbe, 0x13, 0xba, 0xc3, 0xe2, 0x93, 0x94,
    0xea, 0x68, 0x45, 0x9a,
  },
  &case4
};

test_case_t case6 = {
  /* key */
  {
    0x76, 0x10, 0xff, 0x13, 0xd7, 0xbd, 0x68, 0x4b,
    0xcb, 0x87, 0x51, 0x90, 0xb5, 0x4d, 0x7c, 0x4f,
    0x60, 0x75, 0x4f, 0x1a, 0x2d, 0xfd, 0x3a, 0xd7,
    0x10, 0xa4, 0x8b, 0x9f, 0x1d, 0x9b, 0x63, 0x02,
    0xe7, 0xc4, 0x37, 0x14, 0xc6, 0x1c, 0xa6, 0xd6,
    0xe6, 0x42, 0xaa, 0xd6, 0x1f, 0xe4, 0x00, 0x29,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in plaintext */
  32,
  /* associated data */
  {
    0xfe, 0x81, 0xb7, 0x13, 0xc3, 0x53, 0x03, 0x6e,
    0x9f, 0xfc, 0x77, 0x8e, 0x69, 0x61, 0x03, 0x35,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0xda, 0xbb, 0x71, 0x9b, 0xa8, 0x3f, 0xa2, 0x29,
    0xdc, 0x7f, 0x90, 0xe8, 0x30, 0x9d, 0xb3, 0x7c,
    0x40, 0x5e, 0x78, 0x14, 0xaf, 0x6c, 0x37, 0xed,
    0xf0, 0xa3, 0x5d, 0x6e, 0xef, 0x89, 0x90, 0x54,
  },
  &case5
};

test_case_t case7 = {
  /* key */
  {
    0xee, 0x83, 0xbb, 0xa4, 0x55, 0x5b, 0xe9, 0xe8,
    0x0f, 0x15, 0xed, 0xeb, 0x5b, 0x43, 0xc5, 0x31,
    0x41, 0xd4, 0xa5, 0x9c, 0x9e, 0x6c, 0x92, 0xce,
    0x12, 0x3e, 0x3f, 0xd3, 0x1d, 0x5d, 0xc3, 0xd7,
    0x34, 0xa9, 0x06, 0xe1, 0x1c, 0xe9, 0x1f, 0x64,
    0x98, 0x33, 0x62, 0x92, 0x86, 0x0b, 0xd4, 0x4e,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x67, 0x0b, 0xd1, 0xae, 0xa0, 0x12, 0xd2, 0x05,
    0x9a, 0xbb, 0xe1, 0x97, 0x36, 0x23, 0x4d, 0x61,
    0x0d, 0xe2, 0x0b, 0x94, 0x38, 0xd8, 0x73, 0x34,
    0x25, 0x7c, 0x64, 0xdc, 0x07, 0xb6, 0x21, 0xc4,
    0xe8, 0x88, 0xcf, 0x19, 0xc5, 0x4f, 0x0e, 0xb4,
    0x14, 0xaf, 0xad, 0x0f, 0x1f, 0xf2, 0x40, 0x93,
    0x5b, 0xe0, 0x98, 0x96, 0x3e, 0x2f, 0xda, 0x3f,
    0x80, 0x57, 0x55, 0x3a, 0xb8, 0x2e, 0x2e, 0xe0,
    0x60, 0xdd, 0xe5, 0x5b, 0xcc, 0xef, 0x12, 0xf7,
    0x91, 0x2b, 0x24, 0xd7, 0xb6, 0x73, 0x88, 0x1e,
    0x7e, 0xc7, 0x84, 0x15, 0xe5, 0xf4, 0xbe, 0x78,
    0x66, 0xf6, 0xe3, 0x9f, 0x2d, 0x2d, 0xc7, 0x81,
    0x28, 0xe1, 0x62, 0x57, 0x80, 0x2f, 0x9f, 0xc9,
    0x69, 0x01, 0x66, 0xc5, 0xb0, 0x48, 0x05, 0x18,
    0x43, 0x1d, 0xa9, 0x8c, 0x9d, 0x75, 0x74, 0x81,
    0x67, 0x9a, 0x39, 0xfd, 0xa6, 0xff, 0x86, 0xce,
    0xdb, 0x60, 0x32, 0xe9, 0xf2, 0x41, 0x82, 0x04,
    0xc6, 0x83, 0xc7, 0xe1, 0x4b, 0xfa, 0x61, 0xa3,
    0x5d, 0x09, 0xc8, 0xee, 0x27, 0xc6, 0xcd, 0x16,
    0xb4, 0x19, 0xc6, 0x13, 0x69, 0x90, 0x90, 0x2c,
    0x93, 0x87, 0x51, 0xd9, 0xcd, 0x68, 0xce, 0x0a,
```

```
    0x87, 0x35, 0xe0, 0x5e, 0x47, 0x0b, 0x1f, 0xa1,
    0x15, 0x52, 0x11, 0xb4, 0x36, 0xf2, 0xa4, 0xe0,
    0xe5, 0x50, 0x1b, 0x81, 0xce, 0xa9, 0xd6, 0x5e,
    0x71, 0xb3, 0x60, 0x79, 0x0e, 0x7e, 0xd1, 0x06,
    0x24, 0xf9, 0xcb, 0xb9, 0x4f, 0xe4, 0x78, 0x17,
    0x97, 0xaf, 0xa6, 0x5e, 0xee, 0xd8, 0x7c, 0x09,
    0x25, 0xc9, 0xe6, 0xe0, 0xbb, 0x7f, 0x70, 0xdf,
    0x43, 0x65, 0x6e, 0x7a, 0xfb, 0x4d, 0x26, 0xd0,
    0x05, 0xf9, 0x2d, 0x60, 0x86, 0x71, 0x34, 0x27,
    0xce, 0xff, 0x68, 0xba, 0x68, 0xfa, 0xb6, 0x8a,
    0x5c, 0x65, 0x0e, 0x7e, 0xc5, 0x70, 0x26, 0x48,
    0x9f, 0x5a, 0xc5, 0x65, 0x9b, 0xab, 0xb7, 0x49,
    0x70, 0x71, 0xfe, 0x9c, 0x54, 0x7f, 0x17, 0x1d,
    0xe3, 0x70, 0xe3, 0x60, 0x8d, 0x46, 0xba, 0xfd,
    0xc9, 0xf2, 0xf1, 0xa4, 0x96, 0xe6, 0x5d, 0x4a,
    0x62, 0x46, 0xdf, 0x3c, 0x8d, 0x6e, 0x6c, 0xeb,
    0x81, 0x3c, 0x98, 0x11, 0xee, 0x18, 0xaf, 0x4c,
    0x84, 0x15, 0xc6, 0x9f, 0x3d, 0xa7, 0x21, 0x53,
    0x78, 0xac, 0xf6, 0x47, 0xab, 0x38, 0x8a, 0xc3,
    0x5e, 0xde, 0x5a, 0xe9, 0x6d, 0x85, 0xff, 0xbc,
    0x55, 0x86, 0xd8, 0x90, 0x9f, 0x1f, 0x73, 0xca,
    0x71, 0x3c, 0xfb, 0x41, 0x58, 0x70, 0xfe, 0xee,
    0x72, 0xb2, 0x78, 0x55, 0x5d, 0xfc, 0x5b, 0xed,
    0x87, 0xf1, 0x97, 0x0c, 0xda, 0xea, 0x2b, 0x94,
    0x9b, 0xea, 0x1c, 0x22, 0xfe, 0x66, 0x6e, 0xe8,
    0xd1, 0x77, 0x5a, 0x11, 0xe5, 0xb0, 0xe9, 0xaf,
    0x29, 0x3e, 0xa0, 0xb5, 0xa4, 0xc9, 0xf5, 0x0a,
    0xa6, 0x48, 0xdf, 0x64, 0xaf, 0xb4, 0x45, 0x13,
    0x3d, 0x96, 0x49, 0xf6, 0xfb, 0x79, 0xa3, 0x90,
    0x5f, 0xd7, 0x02, 0xcf, 0x81, 0x64, 0x24, 0xf2,
    0x64, 0x8f, 0x73, 0x31, 0xe6, 0x9a, 0xf2, 0x8e,
    0xf4, 0xae, 0xb6, 0x6e, 0x6e, 0x25, 0x1f, 0x01,
    0xc9, 0x31, 0x3b, 0x7e, 0xb5, 0xb4, 0xf7, 0x6a,
    0x8f, 0xa7, 0x38, 0xf1, 0x37, 0x3c, 0x14, 0x0d,
    0xaf, 0xf4, 0x71, 0x5b, 0xb4, 0x29, 0xf2, 0xe9,
    0xcd, 0x1f, 0x72, 0x9a, 0xed, 0x8a, 0xf6, 0xa9,
    0xab, 0x09, 0x27, 0x5e, 0x43, 0xeb, 0xee, 0xae,
    0xf0, 0x5f, 0x93, 0xb2, 0xd8, 0x06, 0x73, 0xb9,
    0x7f, 0x08, 0x71, 0xeb, 0x53, 0xa1, 0x03, 0x72,
    0x46, 0x3a, 0x10, 0x85, 0xc1, 0xa3, 0x05, 0x67,
    0x08, 0x4b, 0x4b, 0x58, 0xfb, 0x8e, 0x53, 0x91,
    0x00, 0xfe, 0x66, 0xcd, 0x2c, 0x2a, 0xd0, 0xbd,
    0x0c, 0xd9, 0x7d, 0xf3, 0x3c, 0x10, 0xd9, 0x4c,
  },
  /* bytes in plaintext */
  512,
  /* associated data */
  {
    0x5f, 0x18, 0xe3, 0x6d, 0xee, 0xb7, 0xaf, 0x7d,
    0x8e, 0xeb, 0x27, 0x54, 0xc5, 0xf7, 0xd1, 0x34,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x8b, 0x3a, 0xc4, 0x22, 0x3c, 0x54, 0xd6, 0x0c,
    0xbc, 0x6b, 0xc8, 0x3f, 0x76, 0x64, 0x0a, 0xee,
```

```
    0xb7, 0xa3, 0xe6, 0x0b, 0xe7, 0xaf, 0x3f, 0x51,
    0x7e, 0x45, 0x8f, 0xe7, 0x73, 0x19, 0x81, 0xd9,
    0x1a, 0x4b, 0xf7, 0xd0, 0x21, 0x1a, 0xb4, 0xf4,
    0xaf, 0x64, 0x83, 0x3c, 0x40, 0x9f, 0x5f, 0xa3,
    0x34, 0xa5, 0x79, 0xae, 0x31, 0xf8, 0xb4, 0x70,
    0x63, 0x9f, 0x25, 0x3b, 0x80, 0x07, 0x9d, 0xde,
    0x12, 0xf9, 0x56, 0xd9, 0xd3, 0xa6, 0x7a, 0xc5,
    0x5b, 0xca, 0x50, 0xc0, 0xa0, 0x11, 0xe4, 0x3f,
    0x56, 0xc1, 0x24, 0x2f, 0x8f, 0x09, 0x16, 0x24,
    0x11, 0x82, 0x34, 0x04, 0x18, 0x7f, 0x13, 0x72,
    0x62, 0x79, 0x42, 0x90, 0x2f, 0x15, 0xdc, 0xd5,
    0x6f, 0xfb, 0xf9, 0xa0, 0xd7, 0x8c, 0xa7, 0xa6,
    0xdb, 0x5b, 0x5b, 0x1d, 0x44, 0x65, 0xb1, 0x86,
    0xd7, 0xc8, 0x1f, 0x6d, 0x59, 0x49, 0x9b, 0xe3,
    0x7f, 0xd9, 0xd8, 0xaf, 0x36, 0xb1, 0x17, 0x57,
    0x5b, 0x70, 0xc7, 0x8c, 0x79, 0x72, 0xc8, 0xd9,
    0xe6, 0x6d, 0x06, 0x8f, 0x49, 0x8c, 0xdc, 0x54,
    0x24, 0xda, 0xc8, 0x66, 0x95, 0x81, 0x7a, 0x87,
    0xcb, 0x65, 0x12, 0x81, 0x78, 0x80, 0xa2, 0x9f,
    0xc0, 0xbe, 0xa8, 0xa7, 0x52, 0xc9, 0x74, 0xe4,
    0x60, 0x49, 0x58, 0x12, 0x94, 0xe7, 0x79, 0xea,
    0xcb, 0x97, 0xc2, 0x21, 0xef, 0xa0, 0x5c, 0xe2,
    0xb0, 0x3a, 0x2a, 0xc7, 0x54, 0x68, 0x61, 0xab,
    0xca, 0x38, 0xc6, 0xa5, 0xa0, 0x16, 0xf8, 0xc7,
    0x00, 0x03, 0xe1, 0xb1, 0xe5, 0xad, 0xff, 0x2b,
    0x4d, 0x27, 0x5e, 0xdc, 0x74, 0x17, 0x28, 0x72,
    0x7b, 0x24, 0x41, 0xe6, 0xbb, 0xc5, 0xdf, 0x36,
    0xa9, 0x3f, 0xcf, 0xef, 0x32, 0xbc, 0x4b, 0x24,
    0xb0, 0xbe, 0xa8, 0x97, 0x2d, 0x14, 0xf2, 0xba,
    0x0d, 0x3c, 0xc0, 0xeb, 0x97, 0x13, 0x46, 0x50,
    0xaf, 0x3a, 0x29, 0x13, 0xcb, 0xd6, 0x19, 0x12,
    0xdf, 0xcc, 0xb8, 0x21, 0x1d, 0x24, 0x17, 0x8a,
    0xa0, 0x17, 0xf0, 0x34, 0x62, 0x18, 0x2f, 0x15,
    0x4c, 0x48, 0x50, 0x9e, 0xab, 0x2c, 0xd8, 0x41,
    0x8a, 0x96, 0x4e, 0xc8, 0x10, 0x9a, 0xdb, 0xad,
    0x5a, 0xe5, 0xe0, 0x86, 0xdf, 0x86, 0x2a, 0x82,
    0xaa, 0xfb, 0xeb, 0x92, 0xf0, 0x0a, 0xca, 0x01,
    0x36, 0xe0, 0xef, 0x63, 0xf8, 0x64, 0x15, 0xcb,
    0x08, 0xf1, 0x84, 0x9b, 0xc1, 0xd5, 0x7e, 0x92,
    0xb3, 0x67, 0xeb, 0x8d, 0x91, 0x17, 0x62, 0xec,
    0x14, 0xdc, 0x4a, 0x86, 0xb7, 0x51, 0xc9, 0xa1,
    0x50, 0x66, 0x94, 0x43, 0x74, 0x6d, 0x03, 0xcd,
    0x48, 0x06, 0xeb, 0xbe, 0xc0, 0xf2, 0x21, 0x79,
    0xac, 0xad, 0x23, 0x06, 0x0b, 0xca, 0x61, 0xd2,
    0xd9, 0xc4, 0xd6, 0x84, 0x19, 0xe4, 0xbf, 0x33,
    0xe0, 0x21, 0x3c, 0xbe, 0x6a, 0xc0, 0x80, 0xd1,
    0xbf, 0x04, 0x31, 0xd5, 0xe6, 0x1d, 0x34, 0xf8,
    0x13, 0x4e, 0x9a, 0xd7, 0xa9, 0xdf, 0x9a, 0x9a,
    0x5f, 0xed, 0x6a, 0x34, 0x1e, 0xd0, 0x54, 0x0e,
    0x42, 0x7b, 0x79, 0xd9, 0x68, 0x3e, 0xc6, 0x40,
    0xfc, 0x5c, 0xc0, 0x3a, 0x95, 0xd7, 0x60, 0x90,
    0x19, 0x7d, 0x07, 0x1e, 0x1a, 0x12, 0x04, 0xfe,
    0x2f, 0x6d, 0xa9, 0x1c, 0x16, 0xb1, 0xdc, 0x7f,
    0x65, 0x3e, 0x4f, 0x7c, 0x27, 0x60, 0x28, 0x58,
    0x7a, 0x94, 0xc1, 0x14, 0xd4, 0x03, 0x8d, 0x4a,
    0xf8, 0xb8, 0x0f, 0x26, 0x78, 0x95, 0x78, 0x81,
    0x5c, 0xbc, 0x50, 0x9f, 0x1a, 0x43, 0xaa, 0x59,
```

```
    0x33, 0xf5, 0x5d, 0x7d, 0xc9, 0x18, 0x75, 0x13,
    0x1d, 0x64, 0x91, 0x31, 0x00, 0x9f, 0x5f, 0x6e,
    0x66, 0x26, 0xc9, 0x26, 0x57, 0xf9, 0xf3, 0x9f,
    0xa2, 0xef, 0x28, 0xd3, 0x00, 0x9a, 0x14, 0x3f,
    0x23, 0x61, 0x20, 0xd6, 0x08, 0x19, 0x74, 0x75,
  },
  &case6
};

test_case_t case8 = {
  /* key */
  {
    0xe2, 0x3e, 0x51, 0x2c, 0xa7, 0x9b, 0x2b, 0x2a,
    0x22, 0xc7, 0x47, 0x96, 0x15, 0x53, 0x92, 0x1e,
    0x2e, 0x4a, 0x82, 0x7f, 0xd8, 0x0f, 0x93, 0xb7,
    0xb3, 0x4a, 0x2b, 0x7b, 0x7d, 0x9b, 0xb3, 0xaa,
    0x74, 0x55, 0x4a, 0xe2, 0x02, 0xbc, 0x37, 0x3e,
    0x02, 0xe8, 0x73, 0x34, 0x27, 0x42, 0xdb, 0xd6,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f,
    0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
    0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f,
    0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27,
    0x28, 0x29, 0x2a, 0x2b, 0x2c, 0x2d, 0x2e, 0x2f,
    0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
    0x38, 0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f,
    0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47,
    0x48, 0x49, 0x4a, 0x4b, 0x4c, 0x4d, 0x4e, 0x4f,
    0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57,
    0x58, 0x59, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x5f,
    0x60, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67,
    0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, 0x6f,
    0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, 0x77,
    0x78, 0x79, 0x7a, 0x7b, 0x7c, 0x7d, 0x7e, 0x7f,
    0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87,
    0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f,
    0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97,
    0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f,
    0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7,
    0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf,
    0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7,
    0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf,
    0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7,
    0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf,
    0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7,
    0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf,
    0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7,
    0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef,
    0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7,
    0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff,
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f,
```

```
    0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
    0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f,
    0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27,
    0x28, 0x29, 0x2a, 0x2b, 0x2c, 0x2d, 0x2e, 0x2f,
    0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37,
    0x38, 0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f,
    0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47,
    0x48, 0x49, 0x4a, 0x4b, 0x4c, 0x4d, 0x4e, 0x4f,
    0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57,
    0x58, 0x59, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x5f,
    0x60, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67,
    0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, 0x6f,
    0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, 0x77,
    0x78, 0x79, 0x7a, 0x7b, 0x7c, 0x7d, 0x7e, 0x7f,
    0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87,
    0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f,
    0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97,
    0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f,
    0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7,
    0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf,
    0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7,
    0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf,
    0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7,
    0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf,
    0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7,
    0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf,
    0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7,
    0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef,
    0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7,
    0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff,
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
  },
  /* bytes in plaintext */
  520,
  /* associated data */
  {
    0xf7, 0x28, 0xad, 0x63, 0x05, 0x84, 0x24, 0x0d,
    0x7b, 0x41, 0xce, 0xf7, 0x16, 0xd2, 0x52, 0x77,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x84, 0xef, 0xf0, 0xf2, 0x27, 0xa1, 0x7b, 0x94,
    0x4a, 0x8d, 0xe3, 0xcb, 0x4c, 0x9a, 0x3b, 0x5d,
    0x67, 0x33, 0x5e, 0x06, 0xcb, 0x28, 0x18, 0x50,
    0xa3, 0x35, 0x65, 0xec, 0x36, 0x66, 0xf9, 0x49,
    0x66, 0xe4, 0x30, 0x4e, 0x4a, 0x44, 0x96, 0xd1,
    0x8f, 0x9d, 0x1e, 0x57, 0xb6, 0x03, 0x93, 0x1c,
    0xd9, 0xdb, 0x59, 0x39, 0xe4, 0x67, 0xb4, 0x97,
    0x65, 0x5a, 0x1e, 0x13, 0x76, 0xb2, 0xb3, 0x9c,
    0x28, 0x16, 0xcc, 0xe2, 0x8e, 0xce, 0xf2, 0xde,
    0x83, 0x8f, 0x5d, 0x6b, 0x7f, 0x5e, 0x86, 0xa8,
    0xc5, 0xbb, 0xb3, 0xa8, 0x14, 0x78, 0x58, 0xc0,
    0x12, 0x83, 0xbc, 0xf7, 0x7d, 0xed, 0xda, 0x43,
    0x0c, 0x9f, 0xea, 0x9e, 0x34, 0x16, 0x25, 0x6f,
    0xbc, 0xc9, 0x68, 0x7d, 0xf7, 0xf4, 0xcb, 0x20,
```

```
    0xfd, 0x13, 0x4d, 0x33, 0x66, 0xa5, 0x8e, 0x14,
    0xad, 0x43, 0x24, 0x95, 0x66, 0x35, 0x7d, 0x6c,
    0xec, 0xee, 0x0e, 0x1a, 0x47, 0xe3, 0x26, 0x89,
    0x35, 0xbd, 0xdf, 0x4c, 0xcf, 0x0b, 0x24, 0xcb,
    0xd8, 0xfe, 0xfa, 0xe2, 0x6e, 0xe8, 0x1d, 0xa2,
    0x03, 0x45, 0x44, 0xa3, 0x0c, 0xf9, 0x19, 0x29,
    0x99, 0x34, 0xf9, 0xcf, 0xc9, 0xe6, 0x42, 0xbb,
    0xbb, 0x49, 0xea, 0x25, 0xf3, 0x59, 0x1f, 0x87,
    0x91, 0x1a, 0xb4, 0x19, 0xc1, 0xb6, 0x9c, 0x18,
    0x38, 0x0a, 0x62, 0x9b, 0x25, 0xb9, 0x98, 0x9b,
    0x22, 0xc8, 0x68, 0xcd, 0xf3, 0x88, 0xb4, 0x13,
    0x83, 0x89, 0xeb, 0x4d, 0x91, 0x95, 0xe9, 0x58,
    0x6f, 0x96, 0x8f, 0x05, 0x3d, 0x61, 0xe2, 0x18,
    0xc0, 0x61, 0x12, 0xde, 0x5b, 0x6a, 0x51, 0x4e,
    0x5e, 0x63, 0x62, 0x7a, 0xb4, 0x5d, 0x10, 0x12,
    0x86, 0xe6, 0x4b, 0x08, 0x73, 0x82, 0x1c, 0xa8,
    0x72, 0xb3, 0xa6, 0xde, 0xed, 0xba, 0x1b, 0xfb,
    0x6a, 0x31, 0xeb, 0xe6, 0xf8, 0xc8, 0x4c, 0xa7,
    0xe7, 0xec, 0x94, 0x08, 0xc9, 0x34, 0x4a, 0x59,
    0xa4, 0xbf, 0xaa, 0x03, 0x3b, 0x89, 0x84, 0x4b,
    0xf5, 0xd8, 0x1f, 0xf8, 0x8d, 0x8f, 0x1f, 0x37,
    0x82, 0xd7, 0xf0, 0x0b, 0x9c, 0x8f, 0x5b, 0x9b,
    0xb0, 0x1c, 0x69, 0x5a, 0x9c, 0x2f, 0xbf, 0x55,
    0xc2, 0x5e, 0x69, 0xba, 0x65, 0x7d, 0x92, 0x1b,
    0x15, 0x0a, 0x55, 0x32, 0xc5, 0x22, 0xf3, 0xb6,
    0xf3, 0x06, 0xc4, 0xda, 0xd2, 0x36, 0xf2, 0x20,
    0x9a, 0x22, 0xa4, 0x0d, 0x37, 0xe0, 0xfe, 0xfb,
    0x0b, 0x6d, 0x30, 0xfd, 0x08, 0xad, 0x9b, 0x2c,
    0x19, 0x63, 0xcb, 0x6a, 0x47, 0xdf, 0xcb, 0xdd,
    0xf8, 0x09, 0x03, 0x43, 0x14, 0xd4, 0x71, 0xf2,
    0x1d, 0x76, 0x77, 0xd9, 0x5a, 0x12, 0x2f, 0xe2,
    0xb6, 0x08, 0x0a, 0x10, 0x77, 0x8b, 0xd4, 0x86,
    0x37, 0xe9, 0x14, 0x62, 0xf3, 0xb8, 0x06, 0x89,
    0x20, 0x23, 0x09, 0xc7, 0xed, 0x50, 0xfe, 0xd8,
    0xf7, 0xd1, 0x7d, 0x6f, 0xca, 0x83, 0xad, 0x67,
    0x33, 0x34, 0x77, 0x5c, 0xd8, 0xf1, 0xfd, 0x48,
    0x2f, 0xee, 0xbe, 0x33, 0x7a, 0x0d, 0x75, 0x07,
    0x14, 0x29, 0xba, 0x7b, 0xc5, 0xc5, 0x6d, 0x62,
    0x20, 0x59, 0x48, 0x79, 0x9c, 0xa9, 0x84, 0xe4,
    0xea, 0x14, 0x6d, 0x0c, 0x68, 0xe4, 0xa5, 0xa5,
    0x10, 0x99, 0x8d, 0xab, 0xa5, 0xd0, 0x73, 0x76,
    0xa0, 0xba, 0xfd, 0x9f, 0x61, 0xca, 0x96, 0xf6,
    0x6c, 0x1a, 0xd3, 0x3a, 0xf8, 0xac, 0x0e, 0xb5,
    0xec, 0x70, 0x81, 0x8e, 0xbc, 0x6a, 0x31, 0x00,
    0x69, 0x15, 0x72, 0xbf, 0x8b, 0x39, 0x1f, 0xb3,
    0x2c, 0x80, 0xf0, 0x37, 0x45, 0x11, 0x75, 0x56,
    0x09, 0x99, 0x49, 0x39, 0x4a, 0xf7, 0x74, 0x61,
    0xd5, 0x99, 0xac, 0xff, 0x10, 0x54, 0x9c, 0x5b,
    0x54, 0x03, 0x9a, 0x5f, 0xe8, 0xf2, 0x4a, 0xad,
    0x0c, 0x1b, 0xa7, 0xd0, 0x51, 0xbb, 0xdc, 0x15,
    0xf7, 0xf9, 0x58, 0x57, 0x58, 0xb7, 0xdb, 0xf5,
  },
  &case7
};

test_case_t case9 = {
  /* key */
```

```
  {
    0x44, 0xb9, 0x77, 0xc9, 0x48, 0xab, 0x9d, 0x55,
    0x5c, 0xfa, 0x2b, 0xb7, 0x1d, 0xae, 0xba, 0x0e,
    0x7d, 0x72, 0x6a, 0xba, 0x3a, 0xbb, 0x08, 0x2f,
    0xba, 0xb5, 0xb5, 0xcb, 0x39, 0x88, 0x1d, 0xa7,
    0x90, 0x57, 0x6c, 0x20, 0xb3, 0x7b, 0xc6, 0x50,
    0x19, 0xee, 0xd6, 0xdf, 0x20, 0x46, 0x94, 0x26,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x25, 0x63, 0x5f, 0x2e, 0x1b, 0x99, 0x27, 0xc0,
    0x9a, 0xa7, 0xf9, 0x9a, 0xac, 0x1c, 0x01, 0x0b,
    0xa6, 0xb9, 0x75, 0xe7, 0xb8, 0x17, 0x26, 0x71,
    0x49, 0x82, 0x60, 0x38, 0x9f, 0x22, 0x47, 0x83,
    0x48, 0x70, 0x2d, 0x48, 0xf3, 0xb7, 0xdc, 0xcc,
    0x58, 0xee, 0x59, 0x28, 0xaf, 0xf4, 0xc4, 0xe8,
    0xbf, 0xa2, 0xf9, 0x02, 0x52, 0xc8, 0xf4, 0xa5,
    0x82, 0xbd, 0xf1, 0xd7, 0x56, 0x8f, 0x80, 0xf6,
    0x8a, 0x17, 0x5d, 0xff, 0xbe, 0xae, 0x2d, 0x20,
    0xe3, 0x87, 0x11, 0x02, 0x19, 0xd4, 0x65, 0x15,
    0x71, 0x09, 0x2d, 0x78, 0x99, 0x4a, 0x75, 0x03,
    0x83, 0x38, 0x8a, 0x88, 0x47, 0xdc, 0xd3, 0xcf,
    0xff, 0x2e, 0x96, 0x19, 0xf0, 0xf5, 0x01, 0xe8,
    0x9b, 0xde, 0x9c, 0xb4, 0xb7, 0xa0, 0x03, 0xa0,
    0x24, 0xb7, 0x74, 0x27, 0xa6, 0xc9, 0xa9, 0xd3,
    0x94, 0x40, 0x44, 0x1f, 0x2c, 0x36, 0xe0, 0x6b,
    0xa6, 0x27, 0x84, 0x18, 0xb5, 0xad, 0x7f, 0x93,
    0x28, 0x3c, 0x86, 0xce, 0x6d, 0x57, 0x36, 0x72,
    0x56, 0x1a, 0x09, 0x2e, 0x00, 0xcd, 0x4b, 0x19,
    0xf0, 0x09, 0x65, 0xa2, 0x21, 0xed, 0xbd, 0x89,
    0xe4, 0xc8, 0xa6, 0x05, 0xd8, 0xe9, 0xaa, 0xff,
    0xdb, 0xed, 0xfe, 0x4f, 0xa5, 0x67, 0x6d, 0xbc,
    0xef, 0x8b, 0xda, 0x28, 0x9b, 0x72, 0xda, 0xe8,
    0x7f, 0x9f, 0xd9, 0x59, 0xb7, 0x29, 0x90, 0xc6,
    0x37, 0x7a, 0xaf, 0xa0, 0x50, 0x18, 0xdd, 0x76,
    0xf2, 0x66, 0xc6, 0x2e, 0xff, 0x5c, 0x77, 0xf9,
    0x1a, 0x07, 0xb0, 0xce, 0x64, 0x5e, 0xf2, 0x8a,
    0xa2, 0xd3, 0xa8, 0xe0, 0xcc, 0x82, 0x69, 0xbd,
    0x01, 0xf7, 0xde, 0xc3, 0x43, 0x3e, 0xb1, 0x2f,
    0x7c, 0x6b, 0x18, 0x16, 0x05, 0xf2, 0xa2, 0x06,
    0x3d, 0x40, 0x73, 0x6c, 0x47, 0xf0, 0x6b, 0xd6,
    0x2d, 0x8a, 0x77, 0x84, 0x1a, 0x80, 0xd5, 0x93,
    0x7e, 0x7b, 0x9d, 0xec, 0x05, 0xa4, 0x03, 0x47,
    0xed, 0x4c, 0x89, 0x4b, 0xdd, 0xf7, 0x36, 0x1f,
    0xf8, 0x5a, 0x94, 0x22, 0xf2, 0xdd, 0x3f, 0x8e,
    0x49, 0x9c, 0xe7, 0x9e, 0xe6, 0xc2, 0x78, 0x18,
    0x30, 0x42, 0xba, 0x0b, 0x11, 0x5a, 0x72, 0x55,
    0x98, 0x1b, 0xf2, 0x16, 0x0a, 0x6e, 0x44, 0x68,
    0x8c, 0x69, 0x28, 0x2c, 0xd8, 0x8a, 0x0f, 0x69,
    0x1a, 0x7f, 0xeb, 0xe1, 0x26, 0x18, 0x70, 0x51,
    0x96, 0x21, 0x99, 0x72, 0xed, 0x19, 0x6a, 0x0a,
    0x2a, 0xc8, 0xe5, 0x15, 0xaa, 0xc3, 0x07, 0xdb,
    0x44, 0xe7, 0xe1, 0xc3, 0xc2, 0x00, 0x21, 0xe0,
    0xf0, 0x55, 0xe1, 0x66, 0x59, 0xe8, 0xfb, 0x0d,
    0xdb, 0x3f, 0xda, 0xb7, 0x04, 0xf6, 0x13, 0x7a,
```

```
    0x4b, 0x4d, 0x87, 0x42, 0x92, 0x7c, 0x63, 0xac,
    0xbb, 0x7f, 0x40, 0x2a, 0xc9, 0xce, 0xe2, 0x92,
    0x6b, 0xb4, 0xd2, 0xd1, 0x72, 0x2c, 0x99, 0xf8,
    0x3e, 0xc7, 0xec, 0xdc, 0xc8, 0xf8, 0x76, 0x34,
    0xb3, 0x26, 0x73, 0xaf, 0xb4, 0x00, 0x84, 0xf2,
    0x71, 0xc3, 0xea, 0xa9, 0x66, 0x99, 0xcd, 0xd3,
    0x5a, 0x6c, 0xdb, 0x55, 0x9c, 0x4a, 0x3c, 0x67,
    0xd0, 0x31, 0xd7, 0x00, 0x11, 0x4d, 0x9a, 0x79,
    0x52, 0xdf, 0x40, 0x8e, 0x5b, 0x01, 0x80, 0x4a,
    0xa9, 0xbe, 0x34, 0xa6, 0x0b, 0x7f, 0xf7, 0x84,
    0xaa, 0x24, 0x2e, 0x74, 0x17, 0x9d, 0x8c, 0x95,
    0x74, 0x43, 0xa8, 0xbf, 0xec, 0x17, 0x14, 0x47,
    0xa9, 0xe8, 0xaa, 0x49, 0x2d, 0x1e, 0xab, 0xb5,
    0xea, 0x44, 0x9a, 0xa2, 0x78, 0xfd, 0xb2, 0x98,
    0x32, 0x6c, 0xd2, 0x78, 0x82, 0xa0, 0x79, 0x74,
    0x85, 0x9f, 0x83, 0xe7, 0x0b, 0x23, 0xe5, 0xbe,
    0x85, 0xf8, 0x40, 0x4c, 0xde, 0x8e, 0xa0, 0x2b,
    0xaa, 0xbc, 0xe4, 0x7a, 0x73, 0xe9, 0x1e, 0x5e,
    0x59, 0x1e, 0xf8, 0x30, 0x5a, 0x2d, 0xe4, 0x1c,
    0x87, 0xa4, 0xc7, 0xa2, 0x81, 0x87, 0xc2, 0x78,
    0xbf, 0x2f, 0xa0, 0xad, 0x04, 0x44, 0x2b, 0x0c,
    0x9d, 0xc6, 0x2a, 0x5b, 0xf3, 0xac, 0xb8, 0x8d,
    0x3e, 0xe9, 0x89, 0x8c, 0xf8, 0x49, 0x7a, 0x20,
    0x09, 0x0d, 0x83, 0x22, 0xdf, 0x65, 0x85, 0x63,
    0x71, 0x25, 0xdf, 0x8f, 0x5a, 0xcc, 0x7a, 0xcf,
    0x65, 0xcb, 0xd0, 0x2a, 0xa6, 0xc4, 0xfc, 0xad,
    0x12, 0xf3, 0x61, 0x5b, 0x56, 0xb9, 0x3c, 0x38,
    0x96, 0xbb, 0xfd, 0x57, 0xda, 0x9f, 0x98, 0xf1,
    0x4d, 0xaa, 0x86, 0xd8, 0x52, 0xf1, 0x4a, 0xed,
    0xf9, 0xb3, 0x7b, 0x5c, 0xec, 0x7e, 0x4c, 0xc6,
    0xf8, 0xaf, 0xe1, 0xc7, 0xac, 0x04, 0xbe, 0x12,
    0x39, 0x43, 0xe5, 0x6a, 0x49, 0x63, 0xbc, 0xe0,
    0xdf, 0xc0, 0x01, 0xac, 0xf0, 0x1f, 0xb9, 0xec,
    0x27, 0x90, 0x81, 0xe5, 0xaf, 0xf0, 0xb8, 0x9a,
    0xdb, 0x03, 0x07, 0x6a, 0x1a, 0xf1, 0xb9, 0x09,
    0xb6, 0xc2, 0x2a, 0x6d, 0xc0, 0x28, 0x64, 0xf6,
    0x42, 0x7b, 0x53, 0x8b, 0x2d, 0x1f, 0x5b, 0x8d,
    0xa8, 0x48, 0xf4, 0x95, 0xa8, 0x35, 0x05, 0x1e,
    0xac, 0x3d, 0x92, 0x89, 0x39, 0x66, 0x14, 0xe6,
    0x3b, 0x81, 0xf8, 0xe8, 0x5b, 0xaf, 0xd5, 0x80,
    0xec, 0x20, 0x7f, 0x9a, 0x3d, 0xed, 0xf0, 0x6a,
    0x11, 0x84, 0x59, 0x73, 0xe6, 0x70, 0x8f, 0xc2,
    0x01, 0xf7, 0xcd, 0x77, 0x23, 0x50, 0x80, 0xa2,
    0xa7, 0x7d, 0x0e, 0x87, 0x4e, 0x1b, 0x67, 0xff,
    0x13, 0x6d, 0xc8, 0xee, 0x7e, 0x6e, 0x01, 0x46,
    0x4c, 0x82, 0x8a, 0x4f, 0x78, 0x3c, 0x00, 0x8d,
    0x17, 0x25, 0x6b, 0x43, 0x21, 0x9c, 0xd8, 0x9d,
    0x90, 0x81, 0xce, 0x95, 0x22, 0x27, 0xee, 0x34,
    0x4a, 0x09, 0x2d, 0x05, 0x7a, 0xdf, 0x5d, 0xdf,
    0x4b, 0x61, 0x8b, 0x2f, 0xc7, 0xf5, 0x85, 0x7f,
    0x83, 0x6b, 0x25, 0x36, 0x23, 0x22, 0xb7, 0x37,
    0x93, 0x43, 0xec, 0x21, 0x78, 0x71, 0xf6, 0x6c,
    0xab, 0x19, 0x7f, 0x21, 0xc4, 0x0f, 0x45, 0x5f,
    0xac, 0x49, 0x67, 0x51, 0x6d, 0x81, 0x4c, 0x04,
    0x96, 0xe3, 0x86, 0xa5, 0xdc, 0x91, 0xa8, 0x8b,
    0x3c, 0xb4, 0x3d, 0xfa, 0x6f, 0x68, 0x61, 0xad,
    0x2c, 0xbd, 0xbe, 0x20, 0x83, 0x6f, 0x3e, 0xfc,
```

```
    0x48, 0xb1, 0x11, 0xd9, 0xbf, 0x0e, 0xb8, 0xa0,
    0xa8, 0x56, 0x46, 0x69, 0xb2, 0xc5, 0xd6, 0xf5,
    0x1c, 0xf6, 0x86, 0x5d, 0xfe, 0x63, 0x80, 0x2c,
    0x44, 0x75, 0x8e, 0x87, 0x38, 0x98, 0x02, 0x21,
    0xc5, 0x7d, 0xee, 0x8a, 0xf9, 0x4b, 0x35, 0x03,
    0xad, 0x7c, 0x8c, 0xf5, 0x77, 0x2b, 0x4e, 0x51,
    0xeb, 0xd4, 0xc3, 0xfe, 0x3d, 0xab, 0x95, 0xd0,
    0x07, 0xb5, 0xde, 0xe0, 0x31, 0x29, 0xcb, 0xf9,
    0x2f, 0xd9, 0x4e, 0x7c, 0x26, 0x0d, 0xca, 0x3b,
    0x43, 0x92, 0xce, 0x69, 0x3f, 0x74, 0x82, 0x0f,
    0x49, 0xa0, 0x58, 0xb5, 0xee, 0x62, 0xb4, 0x7b,
    0x56, 0x57, 0x6b, 0xfa, 0x6b, 0xd7, 0x5d, 0xa3,
    0x39, 0x93, 0xc6, 0xa8, 0xb6, 0x9b, 0x87, 0xb3,
    0xd0, 0x11, 0x46, 0x92, 0xbe, 0x35, 0xec, 0xcb,
    0xae, 0xb5, 0x67, 0x03, 0x5f, 0x01, 0xe7, 0x55,
    0x8f, 0xa3, 0x46, 0x82, 0x2b, 0x36, 0x35, 0x12,
    0xe9, 0xa4, 0xcd, 0xcf, 0x15, 0x50, 0x3b, 0x29,
    0x91, 0xd8, 0x35, 0xff, 0xd5, 0xb2, 0xed, 0x9b,
    0x89, 0x5a, 0x8f, 0xbf, 0x98, 0x49, 0x2a, 0xa7,
    0x13, 0xc9, 0xd2, 0x05, 0x2f, 0x68, 0x5e, 0xc9,
    0xdc, 0x5d, 0xc0, 0xbd, 0xe5, 0x00, 0x07, 0x38,
    0x7a, 0x21, 0x7c, 0x27, 0x4f, 0x64, 0x5b, 0x35,
    0x1b, 0x99, 0x57, 0xf8, 0x7d, 0x8e, 0xc1, 0xd0,
    0x6f, 0xa5, 0xff, 0xb9, 0x8f, 0xa2, 0x57, 0x6d,
    0xf9, 0x7f, 0x28, 0x3f, 0x3a, 0x42, 0x84, 0xed,
    0xfd, 0x1e, 0x8d, 0xe9, 0x79, 0xa7, 0xfb, 0x54,
    0x3b, 0xbd, 0x47, 0xe2, 0xd0, 0xc5, 0x47, 0x25,
    0x39, 0x9c, 0x3e, 0x74, 0x23, 0xd4, 0xa8, 0xdb,
    0x1f, 0x1b, 0x64, 0x31, 0x55, 0x18, 0x22, 0xf7,
    0xa1, 0x5b, 0x9d, 0x0c, 0xb0, 0x19, 0x7d, 0xa3,
    0x18, 0x83, 0xc1, 0xdc, 0x6c, 0x91, 0xf0, 0xcf,
    0xd8, 0x34, 0xd2, 0xcc, 0x93, 0x50, 0x9a, 0xc4,
    0xdb, 0x7a, 0xb4, 0xb1, 0xd4, 0x38, 0xca, 0x09,
    0x2e, 0x4f, 0x19, 0x15, 0x38, 0xe4, 0xdd, 0x93,
    0x4d, 0x47, 0xe6, 0xc6, 0x54, 0x80, 0xba, 0x94,
    0x53, 0x14, 0x44, 0x45, 0x24, 0xdf, 0xbf, 0x80,
    0x8a, 0xb3, 0x5e, 0x58, 0x05, 0x22, 0x6e, 0x1c,
    0x84, 0xfa, 0x53, 0x7f, 0x64, 0xfd, 0x9c, 0x68,
    0x98, 0x5a, 0x9a, 0xf6, 0x27, 0x5f, 0x17, 0xd6,
    0x99, 0xc8, 0x4f, 0x39, 0x9b, 0x84, 0xa0, 0x33,
    0xde, 0x20, 0x0e, 0x46, 0xc8, 0x52, 0x30, 0x39,
    0xb5, 0x5e, 0x11, 0xb3, 0xd2, 0x16, 0x83, 0x96,
    0x65, 0x39, 0xaa, 0x76, 0xbc, 0x21, 0x21, 0xaf,
    0x13, 0x59, 0x7f, 0xab, 0xf9, 0xf1, 0xbd, 0x7b,
    0x89, 0xac, 0xa6, 0xaa, 0x21, 0x05, 0x69, 0xcd,
    0x1b, 0x42, 0x36, 0x69, 0xe8, 0x13, 0xae, 0xb6,
    0x1d, 0x23, 0xc4, 0x6c, 0x4d, 0x9e, 0xa4, 0xfe,
    0x61, 0x1e, 0x6d, 0x67, 0x11, 0x33, 0x94, 0x7c,
    0xc3, 0x2b, 0x10, 0xef, 0x26, 0x7b, 0xdf, 0x28,
    0x45, 0x5c, 0xde, 0xbd, 0xb4, 0xf5, 0x51, 0x6e,
    0xff, 0xe9, 0x13, 0xe0, 0xda, 0xd2, 0x48, 0x87,
    0x43, 0x14, 0x47, 0xed, 0xc6, 0xb4, 0xd4, 0xb7,
    0x76, 0xee, 0x97, 0xce, 0x47, 0x1f, 0x95, 0xec,
    0x05, 0x83, 0x2a, 0x0c, 0x37, 0x6b, 0x35, 0xb9,
    0xa7, 0xdf, 0xc6, 0xa4, 0x65, 0xc7, 0x1a, 0x6a,
    0x32, 0xf1, 0x8d, 0x99, 0xd9, 0xad, 0x14, 0x0f,
    0xb2, 0xa9, 0xc7, 0x4a, 0x00, 0xbe, 0x77, 0x4c,
```

```
    0x0c, 0x25, 0x7a, 0xaa, 0x1c, 0x06, 0x8e, 0x70,
    0x10, 0x75, 0xc6, 0x38, 0x0a, 0xf6, 0xa5, 0x97,
    0xb2, 0x52, 0x00, 0x44, 0xf9, 0xe5, 0x10, 0xf3,
    0x83, 0x23, 0x66, 0xe7, 0x56, 0xcc, 0x4d, 0xfa,
    0x99, 0x2b, 0xfc, 0xd1, 0x97, 0xc4, 0x89, 0xfe,
    0xd0, 0xe1, 0x1a, 0xfc, 0x8f, 0x7c, 0x45, 0x82,
    0x5a, 0x21, 0xe0, 0xc3, 0x61, 0x2d, 0xe7, 0xf0,
    0x9b, 0xcd, 0x94, 0x2b, 0x3c, 0x63, 0x87, 0xb4,
    0x77, 0x08, 0x6a, 0xf2, 0xf2, 0xe8, 0x44, 0xd4,
    0xa0, 0x19, 0x16, 0x43, 0xcf, 0x00, 0xbb, 0x9a,
    0x15, 0xc5, 0x11, 0x19, 0x26, 0xd0, 0xdf, 0x93,
    0x5f, 0x4a, 0x67, 0x18, 0x41, 0x5b, 0xde, 0x71,
    0x24, 0xf6, 0xd0, 0x02, 0x28, 0x3e, 0xa9, 0xcc,
    0x7b, 0xd5, 0xa7, 0x0b, 0x5a, 0x2b, 0x7c, 0x3a,
    0xac, 0x43, 0x93, 0xa5, 0x33, 0x7e, 0xd4, 0xab,
    0x57, 0x9d, 0x31, 0x06, 0xcd, 0x02, 0xfc, 0x11,
    0xf8, 0x6a, 0x50, 0xb3, 0x29, 0x1f, 0xd7, 0x88,
    0x1a, 0x4c, 0x62, 0xc7, 0x8a, 0x7f, 0xd8, 0x74,
    0xcb, 0x6c, 0xeb, 0xaa, 0x83, 0x2b, 0x76, 0x47,
    0x12, 0xb7, 0x96, 0xf7, 0x1a, 0x16, 0xb0, 0x2c,
    0x2a, 0x7a, 0x46, 0x32, 0x12, 0x1f, 0x17, 0xf1,
    0xde, 0x36, 0xff, 0xa5, 0x52, 0xd5, 0xa6, 0x97,
    0xda, 0x05, 0xd7, 0x60, 0x05, 0x41, 0x9f, 0x1c,
    0x0e, 0xd0, 0xcc, 0x82, 0xe7, 0x74, 0x6d, 0x0e,
    0x89, 0x61, 0x28, 0xfc, 0xa2, 0x9f, 0xb5, 0xca,
    0x17, 0x2d, 0xd7, 0x53, 0xb1, 0xe4, 0xd1, 0xe9,
    0xca, 0x26, 0x72, 0xca, 0x10, 0x25, 0xf5, 0x27,
    0x12, 0x84, 0xaf, 0x40, 0x9e, 0xb5, 0x2e, 0x8d,
    0xda, 0x8d, 0xfd, 0xb1, 0x3c, 0x9a, 0x2c, 0xb7,
    0xe5, 0x0b, 0xe0, 0x57, 0x9a, 0xf6, 0xd2, 0x3a,
    0x82, 0x47, 0xfa, 0xe9, 0x80, 0xb2, 0x4e, 0x46,
    0xe6, 0x71, 0x15, 0x17, 0xa4, 0x28, 0xb6, 0x49,
    0x96, 0xf4, 0x3f, 0x09, 0x65, 0xa7, 0x48, 0xbc,
    0x3b, 0xda, 0x00, 0xcf, 0xfc, 0x6f, 0x9e, 0xe4,
    0xbc, 0x72, 0x4e, 0x7e, 0xc1, 0x1b, 0x53, 0xb4,
    0xbd, 0x8a, 0x36, 0xf9, 0x87, 0x45, 0xe5, 0x28,
    0x52, 0xc2, 0x0d, 0xc7, 0x5d, 0xf0, 0x58, 0x7d,
    0xca, 0x65, 0x62, 0xe1, 0x38, 0x7d, 0x41, 0x8f,
    0x2f, 0x56, 0xda, 0xcd, 0xfd, 0x34, 0xfb, 0x3c,
    0x02, 0xb5, 0xd4, 0x3f, 0x8f, 0xd5, 0xa8, 0x56,
    0x85, 0x09, 0x70, 0x98, 0xbd, 0xcc, 0x96, 0xed,
    0xba, 0x06, 0x79, 0xde, 0xa4, 0xf5, 0x4f, 0x5d,
    0x83, 0x03, 0xcc, 0xee, 0x09, 0xf8, 0xce, 0x08,
    0x84, 0x03, 0x01, 0xbf, 0x14, 0x06, 0xde, 0x7c,
    0xce, 0x86, 0xa6, 0x25, 0xa1, 0x72, 0xe9, 0x92,
    0x18, 0xdb, 0x76, 0x06, 0x28, 0xaf, 0xdd, 0x5a,
    0x11, 0x37, 0x15, 0xda, 0xa0, 0x7b, 0x18, 0x0c,
    0x43, 0x91, 0x50, 0xc6, 0x9a, 0x79, 0x4d, 0x4f,
    0xd3, 0x11, 0xbe, 0x5d, 0xec, 0x87, 0xd2, 0x64,
    0x25, 0xa8, 0x0d, 0x06, 0x76, 0xf5, 0x21, 0x18,
    0x15, 0xe1, 0x93, 0x10, 0x0f, 0xd7, 0x9a, 0x14,
    0xf7, 0xe8, 0xc5, 0x16, 0x8d, 0x28, 0xdb, 0x65,
    0xe9, 0x47, 0x20, 0x47, 0xd7, 0x6f, 0x81, 0x56,
    0x73, 0x8b, 0xce, 0xaa, 0x6d, 0xb5, 0x3b, 0x66,
    0x66, 0xdd, 0x71, 0x67, 0x36, 0xc6, 0x4e, 0x59,
    0xc3, 0x26, 0x1a, 0x44, 0x23, 0xf9, 0x1c, 0x42,
    0xcc, 0xa7, 0xe0, 0xa2, 0xf5, 0x7b, 0x9a, 0x57,
```

```
    0x50, 0x06, 0x2e, 0x59, 0x78, 0x21, 0x8b, 0x48,
    0x98, 0x89, 0x5e, 0x7c, 0xd8, 0x0a, 0x52, 0xe6,
    0x75, 0xfb, 0x21, 0xd5, 0x97, 0xb1, 0x24, 0xee,
    0x33, 0x04, 0xf1, 0x93, 0x23, 0xc0, 0x35, 0xb3,
    0xdb, 0xba, 0xb5, 0xcd, 0x4d, 0x16, 0x0f, 0xd6,
    0x5f, 0x59, 0xca, 0x47, 0x3b, 0xc7, 0xe4, 0xa0,
    0x01, 0xe2, 0x0a, 0xd0, 0x1b, 0xde, 0x61, 0x52,
    0x4c, 0x70, 0x7f, 0x7c, 0x45, 0xbf, 0x5a, 0x88,
    0x23, 0x30, 0x3f, 0x64, 0x42, 0x10, 0x7a, 0x36,
    0x9e, 0xfd, 0xb6, 0x3c, 0x60, 0xb2, 0x0e, 0x0a,
    0xa3, 0x91, 0x85, 0x52, 0x4d, 0xfa, 0x76, 0x32,
    0xc0, 0x27, 0xf0, 0x85, 0xd7, 0xf5, 0xd8, 0x19,
    0xf6, 0x8f, 0xe4, 0x49, 0xe7, 0x0e, 0x10, 0x7d,
    0xfa, 0x3f, 0x45, 0xda, 0xe6, 0x3c, 0x40, 0x6e,
    0xea, 0x28, 0x86, 0x2a, 0x4c, 0xb0, 0x34, 0x86,
    0x83, 0x0d, 0x5e, 0x38, 0x9c, 0x78, 0xbe, 0x77,
    0xcc, 0x28, 0xcf, 0xfb, 0x8f, 0x6f, 0xc6, 0x7e,
    0x95, 0x11, 0x49, 0x63, 0x6f, 0x8a, 0xfb, 0x8a,
    0x63, 0x1b, 0xcf, 0x8a, 0x57, 0x3b, 0xfd, 0x1a,
    0x8e, 0xc2, 0xb4, 0x46, 0xb4, 0x84, 0xfe, 0x9c,
    0xd3, 0x27, 0x15, 0x70, 0x00, 0xf1, 0x75, 0x67,
    0xcc, 0x70, 0x4d, 0x44, 0xb0, 0xf1, 0x14, 0x18,
    0xc9, 0x05, 0xb8, 0x2a, 0x0b, 0x14, 0xd7, 0x4d,
    0x7a, 0x2c, 0x29, 0x47, 0x3a, 0x95, 0x67, 0x1a,
    0xcd, 0x9d, 0x19, 0x88, 0x61, 0xd6, 0x95, 0xeb,
    0xc0, 0x89, 0x6f, 0x20, 0xa2, 0xa6, 0x48, 0xbf,
    0x56, 0xdc, 0xbc, 0xd9, 0x73, 0x57, 0x2b, 0x0d,
    0xc0, 0x8b, 0xcf, 0xee, 0x25, 0x61, 0xf3, 0x81,
    0x6a, 0xf5, 0x1c, 0xf4, 0xee, 0x70, 0xd5, 0xed,
    0x29, 0x17, 0x79, 0x25, 0x2c, 0xf7, 0xc6, 0xc6,
    0xc6, 0xf3, 0x50, 0xf5, 0xf5, 0x02, 0x62, 0xdc,
    0xb7, 0x12, 0x99, 0x2a, 0xc8, 0x16, 0xe3, 0x50,
    0x57, 0x6d, 0xd3, 0xfb, 0x06, 0x24, 0xbf, 0x9a,
    0x2d, 0x66, 0x38, 0x1d, 0xd4, 0x03, 0x71, 0xb1,
    0x9b, 0xc2, 0xfb, 0x41, 0xbb, 0x7f, 0x9a, 0xfd,
    0x93, 0x74, 0xc3, 0xc3, 0xd9, 0x61, 0x49, 0x3c,
    0xe4, 0x0b, 0xaa, 0x85, 0x70, 0xc5, 0x36, 0xd6,
    0x51, 0x77, 0xe6, 0xbb, 0xe3, 0xf3, 0xe0, 0x85,
    0xda, 0x7b, 0x86, 0x45, 0xf9, 0x04, 0x4b, 0xad,
    0xe1, 0x6c, 0xef, 0xf8, 0x4b, 0xb4, 0x22, 0x30,
    0x0a, 0xd5, 0xd7, 0xb2, 0x3c, 0x01, 0xdf, 0x8c,
    0xc5, 0x14, 0x8f, 0x66, 0x99, 0x3b, 0x04, 0x08,
  },
  /* bytes in plaintext */
  2064,
  /* associated data */
  {
    0x3d, 0xa5, 0xf4, 0x9d, 0x10, 0xf8, 0x5b, 0xcf,
    0x77, 0x0f, 0x5e, 0x24, 0x0d, 0xb8, 0x40, 0xd7,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x9b, 0x9a, 0xcd, 0xad, 0xd8, 0xdb, 0x40, 0x0d,
    0x49, 0x89, 0x92, 0xb0, 0x1f, 0xcd, 0x20, 0x1f,
    0xa8, 0x67, 0x45, 0x12, 0xbf, 0x30, 0xb7, 0x94,
```

```
    0x37, 0xf0, 0xbd, 0x5a, 0xf1, 0x7e, 0x1e, 0x99,
    0x45, 0xc8, 0x76, 0xd8, 0xd4, 0xd6, 0xd1, 0x73,
    0x20, 0x6f, 0x85, 0x9a, 0x46, 0xec, 0x26, 0xb1,
    0xdd, 0xa1, 0xf1, 0x18, 0xed, 0xd2, 0x64, 0x7e,
    0x80, 0xc3, 0x07, 0x18, 0x06, 0xe7, 0xf5, 0x24,
    0xff, 0x72, 0xee, 0x0b, 0x34, 0xaa, 0xf0, 0x84,
    0xeb, 0x3c, 0x39, 0x52, 0x95, 0x62, 0x62, 0x89,
    0x87, 0x86, 0xa8, 0xc9, 0x41, 0xaf, 0xf3, 0x6e,
    0x69, 0xf1, 0xd4, 0x68, 0x4d, 0x9b, 0x30, 0x5f,
    0x3b, 0x48, 0x1d, 0xa4, 0x29, 0x2c, 0x4c, 0xc5,
    0x5b, 0xe5, 0x6e, 0xbc, 0xc2, 0xdc, 0x42, 0xc8,
    0x89, 0xa8, 0x59, 0xd8, 0x46, 0xf1, 0x0b, 0x0c,
    0x50, 0x05, 0x60, 0x86, 0x05, 0x35, 0x24, 0x16,
    0xfe, 0x80, 0x81, 0xbc, 0x8c, 0x92, 0x26, 0xca,
    0x99, 0x34, 0x4e, 0xa1, 0xbe, 0x52, 0xf0, 0xb8,
    0xd1, 0xae, 0x0a, 0x67, 0x4c, 0xcf, 0xf7, 0x0d,
    0x12, 0x12, 0x84, 0x57, 0x94, 0xa3, 0x98, 0xae,
    0xd3, 0xf3, 0x94, 0xed, 0xed, 0xe4, 0xa2, 0xb5,
    0x67, 0x07, 0x58, 0x49, 0xc4, 0xb3, 0xcf, 0x5c,
    0x32, 0xdb, 0xcb, 0x58, 0x25, 0x44, 0x44, 0xfa,
    0xbb, 0xf6, 0xd1, 0x52, 0x65, 0xfb, 0x9e, 0x41,
    0xf6, 0x4a, 0xa7, 0xc8, 0x8b, 0x88, 0x23, 0x56,
    0x70, 0x80, 0xc8, 0x3f, 0x9c, 0x97, 0x5a, 0x02,
    0xb2, 0xe4, 0x24, 0xb7, 0x3e, 0x3e, 0x55, 0x54,
    0x3e, 0x60, 0x66, 0x31, 0xbc, 0xe8, 0x84, 0x32,
    0x75, 0x4b, 0xc4, 0x67, 0xf7, 0x59, 0x19, 0x4a,
    0x8e, 0x0f, 0x69, 0xf5, 0x57, 0x0a, 0x9d, 0xdc,
    0xf4, 0x62, 0x17, 0xd3, 0xfc, 0xcc, 0x55, 0x0a,
    0x1b, 0xc2, 0x13, 0x22, 0x2e, 0x96, 0x68, 0xaf,
    0x4f, 0xcf, 0x8b, 0x2c, 0x88, 0xaf, 0x03, 0xfc,
    0x1d, 0x2a, 0x08, 0x01, 0x43, 0xb2, 0x50, 0xed,
    0xd5, 0x73, 0x71, 0x5a, 0x83, 0xe6, 0x29, 0x8c,
    0xbc, 0x3c, 0x70, 0xfb, 0x92, 0xa3, 0x44, 0x94,
    0x06, 0x53, 0x5b, 0x42, 0xf7, 0x97, 0xc4, 0x9a,
    0x27, 0x3c, 0xd8, 0x61, 0x5e, 0x54, 0x72, 0x22,
    0xeb, 0xa0, 0xac, 0x46, 0x69, 0xd1, 0x1c, 0x45,
    0xed, 0x6d, 0x61, 0xc3, 0x31, 0x53, 0xf0, 0x63,
    0x19, 0x46, 0xee, 0xa8, 0x61, 0x45, 0x43, 0x19,
    0x00, 0x1f, 0xc2, 0x27, 0x7b, 0xa9, 0xbf, 0x4b,
    0x6c, 0x4b, 0x8c, 0x4f, 0x57, 0x23, 0x90, 0xd0,
    0xb3, 0x8c, 0xfb, 0xa2, 0xd2, 0x54, 0x91, 0x34,
    0x40, 0xf9, 0xc3, 0x45, 0x8f, 0x63, 0xde, 0x5b,
    0x02, 0x38, 0x08, 0x2c, 0x73, 0x70, 0xe1, 0xd6,
    0x1c, 0xe8, 0xb4, 0x43, 0x4c, 0x29, 0xbd, 0x9f,
    0x6d, 0x17, 0x0e, 0xa1, 0xe9, 0x80, 0xbb, 0x10,
    0x88, 0x21, 0x73, 0x9a, 0x40, 0x86, 0xaf, 0xed,
    0x9c, 0xf6, 0xd1, 0x01, 0x85, 0x72, 0x31, 0x2e,
    0xbb, 0x5a, 0x46, 0x46, 0x69, 0xe6, 0x09, 0xb7,
    0x2d, 0x20, 0x2e, 0x69, 0xb4, 0x5c, 0x9c, 0x42,
    0x61, 0x63, 0x4c, 0xa0, 0x18, 0xb0, 0xcf, 0x39,
    0x07, 0xad, 0xc7, 0xdc, 0x33, 0xf9, 0xbd, 0x6f,
    0xa9, 0xe7, 0x35, 0x60, 0xbc, 0xcf, 0x32, 0x6c,
    0x3d, 0x96, 0x31, 0xd9, 0xdc, 0x55, 0x80, 0x71,
    0x35, 0x43, 0xa9, 0x11, 0xd0, 0xea, 0xba, 0x56,
    0xec, 0x0e, 0xbd, 0x6b, 0x11, 0x52, 0xa7, 0xd6,
    0xdb, 0x1a, 0x3c, 0x80, 0xc5, 0xdd, 0x17, 0xc9,
    0x19, 0x5f, 0x5d, 0x26, 0xac, 0x71, 0x94, 0x6e,
```

```
    0x3c, 0x8e, 0xc0, 0x2a, 0x3f, 0xbf, 0x4a, 0x25,
    0x65, 0x37, 0xd6, 0xa0, 0x89, 0x36, 0x4a, 0xbf,
    0x51, 0x64, 0xaf, 0x4b, 0xeb, 0x8f, 0x33, 0xa5,
    0x91, 0x18, 0x0c, 0xb4, 0x9b, 0x75, 0xb5, 0x63,
    0x69, 0xf9, 0x4a, 0xda, 0xd1, 0xf3, 0x34, 0xf9,
    0x62, 0x7c, 0x57, 0xe3, 0xe1, 0x4b, 0x88, 0x7e,
    0x5e, 0xed, 0xa4, 0x15, 0x35, 0xcb, 0x07, 0xe3,
    0x8c, 0x39, 0xfc, 0x09, 0xd7, 0xaa, 0x26, 0xbc,
    0x82, 0xc2, 0x09, 0x01, 0xfa, 0x9f, 0xde, 0x84,
    0xa7, 0x90, 0x13, 0x39, 0x6b, 0x5a, 0x88, 0x23,
    0x2e, 0x9e, 0xff, 0x66, 0xa1, 0xb8, 0x2a, 0x6e,
    0xe1, 0xad, 0xdf, 0x99, 0x9b, 0x8d, 0x03, 0x9c,
    0x86, 0x68, 0xeb, 0x1d, 0xbb, 0xa3, 0x7f, 0x42,
    0x03, 0x1e, 0x65, 0xc3, 0x86, 0x74, 0x1c, 0xc0,
    0xb6, 0x3d, 0xb1, 0xf4, 0x7e, 0x56, 0x6f, 0xd0,
    0x38, 0x7d, 0x76, 0xc7, 0xc1, 0x9f, 0xf6, 0x4e,
    0xf8, 0x25, 0xaf, 0x4c, 0xd4, 0x28, 0xc6, 0x31,
    0xbb, 0xe3, 0x80, 0x3b, 0xc2, 0x8f, 0x5b, 0x79,
    0x81, 0xbe, 0x15, 0xe1, 0xb0, 0xe7, 0x11, 0x24,
    0x53, 0xfa, 0xb7, 0xc0, 0x6c, 0x3a, 0x02, 0xf6,
    0x46, 0xfa, 0x02, 0xd8, 0x96, 0x41, 0x99, 0x81,
    0x1f, 0x78, 0x36, 0x46, 0xc3, 0x78, 0x45, 0x6b,
    0x25, 0xba, 0x8d, 0x79, 0x0e, 0xc1, 0xa8, 0x3d,
    0x33, 0x9f, 0x66, 0xa3, 0x93, 0x2a, 0x40, 0xf1,
    0xed, 0xbc, 0x83, 0x04, 0xb9, 0x5c, 0xc6, 0xfb,
    0x42, 0x29, 0x6a, 0x04, 0xc2, 0xfe, 0x24, 0x36,
    0x14, 0xa5, 0x3c, 0x4d, 0x84, 0xa2, 0x63, 0x16,
    0xbc, 0x34, 0x6c, 0xa4, 0x30, 0xcc, 0xf3, 0x27,
    0x9a, 0xfc, 0x29, 0xe4, 0x2a, 0x5a, 0x1f, 0x0f,
    0x71, 0xa8, 0x12, 0x62, 0xe1, 0xb8, 0xe4, 0xd5,
    0x3e, 0x97, 0xda, 0x4d, 0x4f, 0x47, 0x49, 0x74,
    0xeb, 0x2f, 0x64, 0x6a, 0xfc, 0x9e, 0xe6, 0x97,
    0x75, 0xb2, 0xe3, 0xff, 0xe0, 0x74, 0x22, 0xc4,
    0x20, 0x32, 0x2e, 0x88, 0xc3, 0x15, 0x3d, 0x29,
    0x9d, 0x39, 0xee, 0x68, 0x79, 0x72, 0xe0, 0x2b,
    0xf3, 0x4c, 0x7a, 0xfc, 0x2f, 0x39, 0xf2, 0xec,
    0xb9, 0xd8, 0x2d, 0xd3, 0xc9, 0x78, 0xc3, 0x52,
    0x4c, 0xf9, 0xae, 0xca, 0x23, 0xbe, 0xbb, 0x54,
    0x28, 0x88, 0x5c, 0x51, 0x9e, 0xd4, 0x75, 0xc1,
    0x1b, 0xe3, 0x86, 0xcb, 0xda, 0xab, 0x8e, 0xa9,
    0x89, 0x6a, 0x93, 0x75, 0x45, 0xce, 0xd7, 0x6a,
    0x9d, 0x11, 0xd1, 0x45, 0xb1, 0x16, 0x2a, 0x6b,
    0x26, 0x21, 0xbb, 0x7a, 0xbc, 0x65, 0xac, 0x46,
    0x40, 0xf7, 0x74, 0x74, 0xc6, 0x32, 0x40, 0x4c,
    0x18, 0x80, 0xc8, 0xf7, 0x90, 0x7a, 0x39, 0x32,
    0xa6, 0xcc, 0xf7, 0x13, 0xd9, 0xba, 0x32, 0x80,
    0x62, 0x33, 0xcd, 0xc8, 0x35, 0x58, 0x6d, 0x5b,
    0x19, 0xa2, 0x1e, 0x53, 0x64, 0xea, 0x77, 0xec,
    0xa2, 0xee, 0xc6, 0x4f, 0x60, 0x10, 0x94, 0x12,
    0xb1, 0xca, 0xd6, 0x7d, 0xef, 0x6d, 0xca, 0xb2,
    0xa8, 0xd8, 0x28, 0x11, 0xb4, 0x35, 0xa6, 0x53,
    0x8e, 0x12, 0x18, 0x3b, 0x8d, 0x86, 0x11, 0x37,
    0x9f, 0xa6, 0x36, 0xfd, 0x7a, 0x50, 0x89, 0xe9,
    0x52, 0xbe, 0xca, 0x1d, 0x31, 0xec, 0xff, 0x46,
    0x4f, 0x90, 0xb9, 0x7e, 0x32, 0xa9, 0x3d, 0xd3,
    0x4c, 0x6a, 0xc0, 0x9a, 0x6d, 0x64, 0x0d, 0xa2,
    0xf6, 0xd5, 0x34, 0xac, 0x79, 0x9d, 0x04, 0xf1,
```

```
    0x5a, 0xf9, 0xe3, 0xb6, 0xb9, 0xda, 0xfd, 0x2b,
    0x16, 0x1a, 0x58, 0x7c, 0x1d, 0xdb, 0x46, 0x40,
    0xd1, 0x69, 0x6c, 0x6b, 0x3b, 0xe1, 0x34, 0x9d,
    0xd9, 0xf2, 0xe4, 0x6d, 0x29, 0x8c, 0x1d, 0xfd,
    0x88, 0xb0, 0x7b, 0xb1, 0x37, 0xdf, 0x74, 0x7a,
    0xe2, 0x95, 0x79, 0x33, 0xab, 0x50, 0x05, 0x10,
    0xfc, 0x83, 0xa3, 0x9a, 0x50, 0x72, 0x3f, 0x79,
    0x61, 0xc9, 0x8c, 0x4a, 0xba, 0xce, 0x56, 0x78,
    0x47, 0xf7, 0xca, 0x42, 0x2d, 0x40, 0x5a, 0x5b,
    0x8b, 0xd0, 0x73, 0xf5, 0x33, 0xfd, 0x5c, 0x58,
    0x7a, 0x34, 0x5c, 0x2e, 0x11, 0x8e, 0xf7, 0xa4,
    0xb9, 0x1f, 0x07, 0x2b, 0xcd, 0x0c, 0x85, 0xcd,
    0x3a, 0x9c, 0xfd, 0x24, 0xa7, 0xda, 0x4f, 0x92,
    0x9a, 0xab, 0x43, 0x50, 0x6e, 0xb5, 0x00, 0xf1,
    0x72, 0xce, 0xe8, 0xce, 0xdd, 0xde, 0x01, 0x22,
    0xe2, 0x0a, 0x15, 0xc7, 0x62, 0x57, 0xd9, 0x25,
    0xa0, 0xe0, 0x48, 0x3e, 0x01, 0x58, 0x13, 0x05,
    0x3f, 0x05, 0xc1, 0x67, 0x44, 0x64, 0x38, 0x4e,
    0xd0, 0x52, 0x87, 0x18, 0x57, 0xfd, 0x2a, 0x9e,
    0x7a, 0x14, 0xd4, 0x76, 0xd5, 0x19, 0xc6, 0x16,
    0xe0, 0xa1, 0x8a, 0x79, 0x7d, 0xe9, 0x1a, 0x64,
    0x99, 0x91, 0xc3, 0xe3, 0x80, 0x80, 0xa2, 0x49,
    0x0d, 0xb3, 0xfd, 0x0e, 0xb3, 0x0d, 0xea, 0x66,
    0x9b, 0x16, 0xde, 0x88, 0xb6, 0x34, 0xfb, 0x33,
    0xd7, 0x76, 0xf1, 0xc2, 0x88, 0x00, 0x7e, 0x07,
    0x2d, 0x75, 0xce, 0xdf, 0x5e, 0x0e, 0x96, 0x7a,
    0x7e, 0x4b, 0x0d, 0x32, 0x02, 0xcb, 0x71, 0x4e,
    0xa9, 0x87, 0x82, 0x3f, 0x62, 0xad, 0x97, 0xf2,
    0x40, 0xaa, 0xa7, 0x9f, 0x1d, 0x36, 0xad, 0x74,
    0xdd, 0x95, 0x85, 0x8c, 0x9b, 0xdb, 0xdf, 0xe1,
    0xdb, 0xbb, 0x70, 0xf6, 0x63, 0xc4, 0xa2, 0xde,
    0x8d, 0x5c, 0xa1, 0x45, 0x2d, 0xe5, 0xc1, 0x91,
    0x38, 0xe3, 0x91, 0x55, 0xca, 0xbd, 0x1e, 0x58,
    0xef, 0xc2, 0xd6, 0xee, 0xf0, 0x13, 0xbd, 0xa6,
    0xf1, 0xe0, 0xf3, 0xfb, 0x4f, 0xce, 0x5c, 0x62,
    0x11, 0xbb, 0xab, 0xdb, 0xc7, 0xbb, 0xf1, 0xd3,
    0x28, 0x6b, 0xf4, 0x98, 0x1f, 0x69, 0x21, 0xc3,
    0x92, 0x08, 0x50, 0x21, 0x0b, 0x03, 0x14, 0xca,
    0xd4, 0xc8, 0x69, 0xe5, 0x53, 0x7f, 0x1c, 0x9c,
    0xd4, 0xdc, 0x57, 0x87, 0x6a, 0x60, 0x5f, 0x53,
    0xb0, 0x01, 0x41, 0x65, 0x4a, 0x3d, 0xe6, 0xf0,
    0x9e, 0x2f, 0x07, 0x2f, 0x0b, 0x69, 0x16, 0xe0,
    0x2e, 0x66, 0x87, 0xae, 0xd1, 0x77, 0x72, 0x40,
    0xc4, 0x4b, 0x65, 0x71, 0xe7, 0xbe, 0xcb, 0x1c,
    0x05, 0x43, 0x8c, 0x08, 0xe6, 0xac, 0x63, 0x16,
    0xa6, 0x89, 0x8f, 0xc2, 0xa1, 0x8e, 0xec, 0xa3,
    0x13, 0x5d, 0x6b, 0x1d, 0x10, 0x90, 0xbd, 0xed,
    0xc9, 0x45, 0x19, 0x11, 0xa3, 0x2c, 0x3c, 0x99,
    0xb6, 0x0f, 0x4c, 0xe4, 0xb3, 0xcc, 0x20, 0xa5,
    0xd5, 0xdd, 0x3a, 0xcd, 0x36, 0x48, 0xb7, 0x73,
    0x41, 0xfc, 0x5a, 0xcb, 0x02, 0x83, 0x3b, 0xb3,
    0x8a, 0x72, 0x75, 0x5e, 0xfc, 0x54, 0x0b, 0xa1,
    0x0c, 0xd7, 0xea, 0x17, 0x3b, 0x7e, 0x7e, 0x9e,
    0xbe, 0xe4, 0x41, 0x41, 0x47, 0xb2, 0x02, 0xac,
    0x09, 0x9a, 0x13, 0xfb, 0xac, 0xdd, 0x9b, 0x73,
    0xc2, 0x14, 0x01, 0x02, 0x86, 0x79, 0xf8, 0xc0,
    0xc2, 0x85, 0x8d, 0x43, 0xc8, 0x6d, 0x6b, 0x7b,
```

```
    0xcc, 0xc3, 0x74, 0xde, 0xea, 0x09, 0x9e, 0x65,
    0xc1, 0x3e, 0x1f, 0xf2, 0xa9, 0x1a, 0x65, 0xbe,
    0x0d, 0x2e, 0xd5, 0x0e, 0xe2, 0x35, 0x4a, 0x9b,
    0x8a, 0xd6, 0x72, 0x6b, 0x3e, 0x23, 0xe2, 0x2a,
    0x9d, 0x0b, 0x57, 0xe1, 0x33, 0x14, 0x6c, 0xcf,
    0xea, 0x56, 0xec, 0x0b, 0x4d, 0x05, 0x7c, 0x50,
    0xee, 0x49, 0x4d, 0xfa, 0x3a, 0xd9, 0x1e, 0x6b,
    0x0d, 0xc6, 0x33, 0xf0, 0x65, 0x78, 0xf4, 0x51,
    0x58, 0xa7, 0x04, 0xc5, 0x78, 0x03, 0x28, 0xad,
    0x10, 0x2e, 0x5c, 0x79, 0x26, 0xeb, 0x1f, 0xd4,
    0xc0, 0x43, 0x92, 0x38, 0xf3, 0xa7, 0xb5, 0x1e,
    0x3e, 0xe8, 0xe7, 0xc3, 0x77, 0xcf, 0x83, 0xb5,
    0xbf, 0x32, 0x8e, 0x57, 0xec, 0xc7, 0x46, 0x8e,
    0x35, 0xba, 0x4e, 0x7c, 0x28, 0xd6, 0x21, 0xb6,
    0x3e, 0x63, 0x86, 0x9a, 0xb2, 0x64, 0x5d, 0xb6,
    0xfc, 0xdd, 0xe3, 0x89, 0x4d, 0xbd, 0x6d, 0x16,
    0x17, 0x7f, 0xce, 0x6d, 0x3d, 0x57, 0x6d, 0xc3,
    0x8f, 0xdd, 0x4e, 0x5c, 0xa7, 0xfe, 0x31, 0x58,
    0x5b, 0x21, 0xe1, 0xd1, 0xf3, 0xb7, 0x3b, 0x05,
    0x68, 0x40, 0xad, 0x3e, 0x05, 0x95, 0xca, 0x1e,
    0xdd, 0x0e, 0x08, 0xe7, 0xac, 0xe6, 0x3f, 0x92,
    0x2a, 0x28, 0xc0, 0xef, 0x65, 0x9b, 0xf8, 0xfb,
    0x20, 0x51, 0xd7, 0x25, 0x97, 0xcf, 0x24, 0x7a,
    0x06, 0x3d, 0xac, 0xdc, 0x10, 0x37, 0xfa, 0xfb,
    0x1c, 0xd4, 0xdf, 0xd4, 0xa1, 0xd3, 0x42, 0xa5,
    0x9c, 0x4a, 0xcd, 0xc2, 0xd1, 0xf2, 0xb4, 0x52,
    0xa9, 0xe8, 0x08, 0xfa, 0x3a, 0x6d, 0x06, 0xd9,
    0x28, 0x50, 0xee, 0x7c, 0x0c, 0xeb, 0x98, 0x2d,
    0xda, 0x70, 0x4c, 0x6f, 0xab, 0x06, 0x53, 0xf8,
    0xea, 0xc4, 0x8b, 0xb4, 0x5b, 0x69, 0x4a, 0xb9,
    0xcb, 0xe7, 0x38, 0xc4, 0x5b, 0x15, 0x6a, 0x97,
    0xfd, 0x0d, 0x0f, 0xa1, 0x7e, 0xe2, 0x7b, 0x28,
    0x08, 0xcd, 0x51, 0x48, 0x39, 0xeb, 0xb4, 0x57,
    0x48, 0x1b, 0x6b, 0x28, 0x4d, 0x86, 0x44, 0xa3,
    0x68, 0x5e, 0x31, 0x8a, 0x3c, 0x20, 0xc4, 0x40,
    0x8a, 0xae, 0xd0, 0xd0, 0x8f, 0x93, 0xae, 0x89,
    0x21, 0xa9, 0xc6, 0x4b, 0x45, 0xfc, 0x9a, 0xfe,
    0x09, 0x6f, 0x15, 0x07, 0xc8, 0x94, 0x14, 0xb6,
    0xbd, 0x7e, 0x16, 0x0e, 0x50, 0xf8, 0xa9, 0x78,
    0x95, 0x9b, 0x7e, 0xf5, 0xb5, 0x7d, 0xfd, 0xa1,
    0x98, 0x26, 0x52, 0x17, 0xdb, 0xfc, 0xe4, 0x59,
    0xaf, 0x06, 0xe2, 0x06, 0x22, 0x23, 0x09, 0x4b,
    0x40, 0x8d, 0x43, 0x03, 0xd0, 0x88, 0x95, 0xee,
    0x9c, 0x9f, 0xe6, 0x01, 0x0a, 0x1e, 0x43, 0x5b,
    0xd5, 0xc5, 0x29, 0x55, 0x52, 0x0e, 0xe9, 0xb2,
    0x3b, 0x49, 0xf4, 0x85, 0xd7, 0xc5, 0xd1, 0x39,
    0x16, 0x96, 0x4d, 0x58, 0x3b, 0x45, 0x4a, 0xe0,
    0x00, 0xad, 0xdb, 0x01, 0x14, 0x42, 0x78, 0x80,
    0x27, 0x2a, 0x2e, 0x97, 0x50, 0x29, 0x12, 0x0e,
    0x38, 0x2f, 0xf9, 0xe3, 0x78, 0xd4, 0x69, 0x29,
    0x9e, 0xfd, 0xdb, 0x2a, 0x5b, 0x39, 0x28, 0x5f,
    0x2f, 0xde, 0xd8, 0x25, 0x75, 0x93, 0x77, 0x6c,
    0x57, 0xc3, 0x58, 0xa6, 0x95, 0xae, 0xa4, 0xd1,
    0xce, 0x61, 0x58, 0x92, 0x9a, 0xe0, 0x6b, 0x3f,
    0xa2, 0xd7, 0x0f, 0x21, 0xf0, 0x3b, 0x4f, 0x68,
    0xfe, 0xcd, 0xca, 0x19, 0xfe, 0xcc, 0xdf, 0x09,
    0x39, 0xce, 0xef, 0x98, 0x59, 0x3d, 0x84, 0xc0,
```

```
    0xf6, 0xb2, 0x9c, 0xb3, 0xbe, 0x08, 0x61, 0xf3,
    0xdb, 0x1f, 0x4e, 0xb6, 0x83, 0xab, 0x11, 0xed,
    0x35, 0xce, 0x6c, 0xfc, 0xb5, 0x1c, 0x17, 0x9d,
    0x39, 0xf5, 0x21, 0xf7, 0x60, 0x66, 0x1b, 0xc4,
    0x52, 0xea, 0x0d, 0x18, 0x16, 0xc2, 0xc6, 0x71,
    0xa7, 0x29, 0x29, 0x82, 0xb1, 0xea, 0x04, 0x1b,
    0x51, 0xac, 0x83, 0x2a, 0x3a, 0xd2, 0x4a, 0x76,
    0x03, 0x2a, 0xdb, 0xc9, 0xd4, 0x80, 0x6e, 0xf8,
    0x86, 0x5e, 0x42, 0x5d, 0xd3, 0xe4, 0x2d, 0xbf,
    0xca, 0x80, 0xd9, 0x8e, 0x5e, 0xeb, 0x48, 0x23,
    0x40, 0x7d, 0x6a, 0xf6, 0x96, 0x50, 0xf3, 0xdf,
    0xf9, 0x11, 0xae, 0x48, 0x7b, 0x56, 0x80, 0xf0,
    0x3b, 0x55, 0x41, 0x9c, 0xcc, 0xeb, 0x52, 0x87,
    0xa2, 0xfe, 0xb7, 0xa7, 0xb1, 0x2d, 0x52, 0x8d,
    0x1f, 0x00, 0x02, 0xa8, 0x27, 0xc7, 0xda, 0xc2,
    0x82, 0xd8, 0x5e, 0x13, 0x9f, 0x3f, 0x03, 0xe5,
    0x5f, 0x2d, 0x05, 0xbd, 0x17, 0x8c, 0x32, 0xcf,
    0xaa, 0x57, 0xcb, 0x42, 0x55, 0x35, 0x50, 0x67,
    0x22, 0xe8, 0xb4, 0x80, 0x62, 0xac, 0x02, 0xe7,
    0xd7, 0xa2, 0x1b, 0x44, 0x99, 0x3a, 0x84, 0x8b,
    0x59, 0x2a, 0x4c, 0x85, 0xac, 0x27, 0xf9, 0x43,
    0x77, 0xb4, 0x27, 0xc4, 0x85, 0xef, 0xfe, 0xf0,
    0x7d, 0x5b, 0xf6, 0xce, 0xc9, 0x0f, 0x20, 0xa0,
    0x81, 0x0f, 0x25, 0x26, 0xb8, 0xd6, 0x0e, 0xc3,
    0x0f, 0x8f, 0x79, 0x4c, 0xf1, 0x54, 0x45, 0x67,
    0xad, 0x34, 0x1f, 0x15, 0xa5, 0x18, 0xd0, 0x8f,
    0x5d, 0xa2, 0xf4, 0x49, 0xf8, 0x90, 0x3a, 0x33,
  },
  &case8
};

test_case_t case10 = {
  /* key */
  {
    0xdb, 0x0d, 0xb2, 0x58, 0x87, 0xd4, 0x7b, 0xdb,
    0x74, 0x93, 0x1a, 0x7f, 0x48, 0xc3, 0x65, 0xcd,
    0x6a, 0x76, 0xc7, 0x9a, 0x26, 0x53, 0xe4, 0xa2,
    0xe1, 0x2b, 0x0d, 0x7e, 0x4c, 0x81, 0x3f, 0x93,
    0x2d, 0xf6, 0xa8, 0xc9, 0x21, 0xb6, 0x2a, 0x2e,
    0x64, 0xaf, 0xa6, 0x9f, 0xc1, 0x99, 0x97, 0x8d,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0xc4, 0xdf, 0x50, 0x73, 0x99, 0x95, 0x30, 0x24,
    0x6d, 0x76, 0x2f, 0x25, 0xa6, 0xe8, 0x0b, 0x37,
    0x78, 0x9c, 0x1b, 0x10, 0x3f, 0xa2, 0x97, 0x68,
    0x55, 0x4e, 0x05, 0x32, 0x0b, 0xa6, 0x66, 0x88,
    0x42, 0x70, 0xc4, 0xda, 0x19, 0x4f, 0xf5, 0x1a,
    0xab, 0x80, 0x6b, 0x59, 0x9f, 0x54, 0x32, 0xb1,
    0x88, 0x39, 0x1f, 0x80, 0x3f, 0x24, 0xc8, 0xd3,
    0x1a, 0x2a, 0x0b, 0x51, 0x67, 0x35, 0xf5, 0x3f,
    0xdb, 0x53, 0xff, 0xdb, 0xfd, 0xaa, 0x57, 0xdb,
    0x1e, 0x82, 0xfb, 0x65, 0x7f, 0xd7, 0xd3, 0x89,
    0x25, 0xbc, 0x3a, 0x72, 0x98, 0x56, 0x9f, 0x3f,
    0x7d, 0xe2, 0x75, 0x76, 0x40, 0xa8, 0x04, 0x88,
```

```
    0xfa, 0x38, 0x6c, 0x30, 0xba, 0x60, 0x4c, 0xec,
    0xf1, 0x96, 0x6f, 0xe3, 0xa4, 0x60, 0x6a, 0xbe,
    0x3a, 0xec, 0x16, 0xb6, 0x76, 0xd6, 0xe9, 0x80,
    0x7e, 0x28, 0x21, 0xd7, 0xd2, 0x0d, 0xbc, 0xa9,
    0xb2, 0x73, 0x8f, 0x7b, 0x6c, 0x92, 0x91, 0xa4,
    0x06, 0xa1, 0xa0, 0x73, 0x8d, 0xdb, 0x8c, 0x1c,
    0xc9, 0x6b, 0x43, 0x00, 0x64, 0x43, 0xf5, 0x27,
    0x62, 0x5c, 0x24, 0xb3, 0x82, 0x7c, 0x7b, 0xc4,
    0x1f, 0x72, 0x10, 0x6a, 0x4a, 0xe4, 0xe3, 0x99,
    0x28, 0x69, 0x57, 0x91, 0x64, 0xa5, 0xa2, 0xa4,
    0x94, 0x38, 0x77, 0xbc, 0xc1, 0xf1, 0x82, 0x00,
    0x41, 0xec, 0x0d, 0x16, 0xde, 0xfc, 0xb1, 0x01,
    0xa0, 0xd5, 0x5c, 0x4c, 0x70, 0xf8, 0xf7, 0xaf,
    0xd3, 0x3c, 0x88, 0xad, 0x63, 0x87, 0xeb, 0xf0,
    0xf6, 0x13, 0xa5, 0xf5, 0x4f, 0xdc, 0x1e, 0xc5,
    0xc3, 0x2f, 0x13, 0x65, 0x0b, 0x3f, 0x19, 0x83,
    0x52, 0xeb, 0x33, 0xa5, 0xfd, 0x9e, 0x0a, 0x14,
    0x33, 0x92, 0xb1, 0x1c, 0x66, 0x2b, 0x3d, 0x11,
    0x66, 0x4f, 0xc7, 0x81, 0x9c, 0x05, 0x8e, 0x05,
    0x81, 0xa2, 0x5c, 0xd3, 0x26, 0x11, 0x26, 0x40,
    0xf4, 0x4e, 0xca, 0x96, 0xd3, 0x36, 0xec, 0xf3,
    0x90, 0xe1, 0x3a, 0xcc, 0x12, 0x23, 0x21, 0x49,
    0x72, 0xa7, 0x43, 0xcd, 0xb0, 0xf1, 0x74, 0x2b,
    0x6e, 0x73, 0x9f, 0x56, 0x1d, 0x0a, 0x15, 0xad,
    0xb5, 0xb1, 0xe8, 0xdf, 0xa1, 0xb1, 0xa9, 0x06,
    0xee, 0x4a, 0xe6, 0xac, 0x07, 0x99, 0x3e, 0x74,
    0x69, 0x78, 0xd3, 0x4a, 0x42, 0xd8, 0xd9, 0x8c,
    0x55, 0x35, 0xf0, 0x83, 0xa2, 0xa3, 0xac, 0xf0,
    0xcd, 0xb6, 0xd7, 0x7b, 0x4a, 0x35, 0x81, 0xf5,
    0x8b, 0x67, 0x9b, 0xc2, 0x03, 0xf4, 0xdc, 0x07,
    0xe0, 0xe6, 0xf2, 0x99, 0xa6, 0x7f, 0x4d, 0xf4,
    0xee, 0xbe, 0x17, 0x14, 0x34, 0xd0, 0xaa, 0x7d,
    0x14, 0x80, 0x27, 0x04, 0x9f, 0x93, 0x53, 0xa6,
    0x2b, 0xde, 0xb6, 0x6d, 0x10, 0x64, 0x92, 0x66,
    0x65, 0x44, 0x1d, 0xec, 0x46, 0xb4, 0xdd, 0x2b,
    0x78, 0x95, 0x84, 0x65, 0xb6, 0xcc, 0x26, 0x36,
    0x09, 0x8e, 0xf1, 0x6d, 0x9c, 0x56, 0xfd, 0xf4,
    0xff, 0x16, 0xa1, 0x95, 0x45, 0x55, 0x94, 0x50,
    0x4c, 0xe1, 0xb2, 0xe8, 0xad, 0x67, 0x0d, 0xd9,
    0x95, 0x76, 0xac, 0xfe, 0x0c, 0xb7, 0x04, 0xe7,
    0x67, 0x81, 0x33, 0xbf, 0xae, 0x50, 0x4e, 0xb5,
    0xef, 0xff, 0x43, 0x41, 0x27, 0xb7, 0x60, 0xff,
    0x33, 0x29, 0x37, 0x0d, 0x49, 0xc9, 0xd8, 0x25,
    0x73, 0xf9, 0x80, 0x3e, 0x1d, 0x86, 0x9c, 0xed,
    0xa6, 0x41, 0x77, 0x80, 0x9f, 0x44, 0x7e, 0x7d,
    0x07, 0x91, 0x33, 0x87, 0xa2, 0x5e, 0xa4, 0x9c,
    0xc1, 0x84, 0xba, 0x4a, 0x0a, 0x70, 0xf2, 0x05,
    0x9d, 0x5a, 0xea, 0x6e, 0xe7, 0xbd, 0x66, 0x59,
    0xe5, 0xc0, 0x09, 0x52, 0x97, 0x54, 0x9d, 0x64,
    0x7e, 0xb0, 0xf9, 0xd3, 0xc5, 0xd7, 0xe3, 0x53,
    0x09, 0x37, 0xaf, 0x72, 0x6d, 0x50, 0x64, 0x03,
    0x26, 0x28, 0x27, 0x64, 0x91, 0x43, 0x59, 0x10,
    0x2b, 0x61, 0x8c, 0xe4, 0x05, 0xc3, 0xf0, 0xce,
    0xb7, 0xf0, 0x4a, 0x93, 0x47, 0xb5, 0xcf, 0x44,
    0xf4, 0xee, 0xb9, 0xbe, 0x18, 0xfe, 0xe2, 0xb9,
    0xe0, 0x98, 0xa5, 0x30, 0x0d, 0x3c, 0x43, 0xa5,
    0xd2, 0x4b, 0xf0, 0x2b, 0x21, 0x8a, 0x25, 0xd9,
```

```
    0x50, 0xc0, 0x91, 0xc2, 0x5b, 0xd1, 0x20, 0xac,
    0xbe, 0xd3, 0xd1, 0x0c, 0x45, 0x38, 0x4e, 0x8d,
    0xe3, 0xfa, 0x60, 0xbf, 0xac, 0x96, 0x99, 0x1f,
    0x9d, 0x1e, 0x10, 0x11, 0x79, 0xb9, 0x75, 0x24,
    0xf9, 0x84, 0x48, 0x0c, 0x15, 0xb0, 0xaf, 0x94,
    0x06, 0x28, 0xdc, 0x72, 0x6a, 0xea, 0x8d, 0xa4,
    0x2a, 0x6e, 0xf9, 0x9d, 0x7e, 0x74, 0x37, 0xa1,
    0xd3, 0x75, 0x42, 0x2d, 0x44, 0x44, 0xc5, 0xef,
    0x7a, 0x4d, 0xe4, 0xbd, 0xe7, 0x51, 0x24, 0xa2,
    0x5f, 0x03, 0x21, 0xdc, 0x4c, 0xdc, 0xb4, 0x98,
    0x70, 0x35, 0xe3, 0x6d, 0xf2, 0xbc, 0x4e, 0x87,
    0x43, 0x2a, 0x60, 0x3c, 0x39, 0xae, 0x53, 0x8d,
    0xb0, 0xa6, 0x12, 0x54, 0x23, 0x80, 0x15, 0x3b,
    0x42, 0x2c, 0x93, 0x65, 0x45, 0x7a, 0xbd, 0x1c,
    0xdc, 0xf1, 0xb8, 0x5f, 0x86, 0xab, 0xcb, 0x23,
    0xe3, 0xf2, 0x1c, 0x50, 0x84, 0x19, 0x3e, 0x5c,
    0xc9, 0x2a, 0x5d, 0x36, 0x13, 0x96, 0xc1, 0x5c,
    0xc5, 0x1b, 0x1b, 0x9c, 0xaa, 0x6f, 0x93, 0xe8,
    0xd1, 0x16, 0x4e, 0x57, 0x39, 0x1f, 0x94, 0x4b,
    0x15, 0x22, 0xf4, 0xe1, 0xbd, 0xd2, 0xf0, 0x58,
    0xdc, 0xfa, 0x5d, 0x44, 0x1b, 0xa1, 0x25, 0xc4,
    0xc9, 0xa4, 0xf1, 0xeb, 0x29, 0x84, 0x8b, 0x01,
    0x2a, 0x15, 0x97, 0x3d, 0x23, 0x36, 0x00, 0xd2,
    0xb1, 0xee, 0xff, 0xf3, 0x06, 0x6d, 0xa8, 0x22,
    0xc3, 0x53, 0xa0, 0x21, 0x5f, 0x78, 0xc3, 0x32,
    0x7b, 0xba, 0x0f, 0x0e, 0xfb, 0xec, 0x15, 0x9f,
    0x67, 0x4b, 0x18, 0xf4, 0xac, 0xe5, 0x4d, 0x9e,
    0xbd, 0x8b, 0xd1, 0x35, 0xae, 0xbf, 0x5b, 0xe8,
    0x34, 0x15, 0x1b, 0xc6, 0x4d, 0x7d, 0x24, 0x2d,
    0x80, 0x42, 0x21, 0xa0, 0xf8, 0xb8, 0x06, 0x50,
    0x6e, 0x40, 0xed, 0xf7, 0x85, 0xcc, 0x3a, 0x71,
    0x8a, 0x5c, 0x42, 0x33, 0x75, 0xb8, 0x74, 0x88,
    0xdb, 0x2a, 0x91, 0x6d, 0x56, 0xa6, 0x33, 0x72,
    0x0d, 0xe3, 0xf5, 0x3e, 0x86, 0xff, 0xc5, 0xcb,
    0xe6, 0xb6, 0x13, 0x82, 0x70, 0xe6, 0x66, 0xd1,
    0xb1, 0x55, 0x5d, 0x6b, 0x3f, 0x35, 0xcd, 0x3d,
    0xf6, 0x0f, 0xaf, 0x78, 0xaa, 0x8a, 0xbd, 0x3d,
    0x8d, 0x7b, 0xa3, 0xe9, 0x19, 0x8e, 0x42, 0x50,
    0x2c, 0xa9, 0x64, 0x1c, 0x7e, 0x78, 0x48, 0x1b,
    0xa1, 0xfa, 0x3e, 0x51, 0x66, 0xca, 0x8e, 0x47,
    0xae, 0x9c, 0x65, 0x73, 0x45, 0x8e, 0xfa, 0xa4,
    0xcb, 0x33, 0x4f, 0x47, 0xb3, 0x24, 0x16, 0x0f,
    0x45, 0x2e, 0x89, 0x77, 0xcf, 0xae, 0x63, 0x15,
    0xcb, 0x24, 0xd6, 0x01, 0x59, 0x7a, 0x7d, 0x40,
    0xef, 0x1d, 0x67, 0xdb, 0xd8, 0x4a, 0x13, 0x6f,
    0xe4, 0x69, 0xa2, 0x6c, 0x13, 0x27, 0x01, 0x64,
    0x69, 0x25, 0xfa, 0xdf, 0x02, 0x66, 0x8f, 0x0e,
    0x2c, 0xae, 0xaf, 0x46, 0xcb, 0x90, 0x38, 0x37,
    0x1b, 0x57, 0xf2, 0x06, 0xfb, 0x47, 0x11, 0x77,
    0xc0, 0x4f, 0x0c, 0x4e, 0xcd, 0x8f, 0x49, 0xf8,
    0x58, 0xd9, 0xc9, 0x08, 0x26, 0xe9, 0xd0, 0xcb,
    0x43, 0xa5, 0xca, 0xa9, 0x39, 0x9a, 0x4d, 0x18,
    0x9e, 0x1b, 0x94, 0x0d, 0xc9, 0xc4, 0x77, 0xf7,
    0x23, 0x3e, 0xa3, 0x93, 0xca, 0x37, 0xb1, 0x57,
    0x92, 0x88, 0x76, 0x90, 0x98, 0x47, 0xc6, 0x05,
    0x04, 0x22, 0x6d, 0x4c, 0x0a, 0xdb, 0xe1, 0x0f,
    0x40, 0x3e, 0x98, 0x24, 0x7b, 0xbb, 0xab, 0xb9,
```

```
    0xec, 0x33, 0xec, 0xaf, 0x62, 0xaa, 0x8d, 0xf4,
    0x89, 0x1e, 0x8a, 0xd8, 0x54, 0xff, 0x3f, 0x6a,
    0x7c, 0x4d, 0xc8, 0xf0, 0x3f, 0x7b, 0xce, 0x68,
    0xdf, 0x47, 0x3e, 0x8f, 0x28, 0xa4, 0xd8, 0xb5,
    0x9a, 0xaa, 0xfa, 0x1a, 0x18, 0x2b, 0x05, 0xd8,
    0xca, 0x11, 0xa6, 0xf1, 0xf5, 0xce, 0xfd, 0xef,
    0x1f, 0x58, 0x63, 0xcb, 0xb4, 0xd5, 0x78, 0x51,
    0xf9, 0xd6, 0x8d, 0x93, 0xd2, 0x44, 0x10, 0x47,
    0x80, 0x29, 0x95, 0xd3, 0x95, 0xa8, 0xe2, 0x49,
    0x77, 0xcd, 0x36, 0xb4, 0xec, 0x2f, 0x7a, 0xd9,
    0x45, 0x26, 0xb6, 0x14, 0x03, 0x8b, 0xe9, 0x4d,
    0x18, 0xed, 0xe6, 0xb9, 0xe5, 0x95, 0x47, 0xa5,
    0x7d, 0x11, 0x3b, 0x0e, 0x84, 0x06, 0xfa, 0xca,
    0xb4, 0x0e, 0x8a, 0xfc, 0xab, 0x9b, 0x11, 0xb9,
    0x00, 0xa6, 0x74, 0xbb, 0x70, 0x9d, 0x44, 0x30,
    0xa8, 0x5c, 0xa1, 0xfa, 0xd0, 0x39, 0x7d, 0x77,
    0x4a, 0xd4, 0xe4, 0x89, 0xd2, 0xda, 0x8b, 0x58,
    0x93, 0x7f, 0xb7, 0xf9, 0x42, 0xcb, 0x5a, 0xec,
    0xeb, 0xd8, 0x6e, 0x45, 0x4f, 0x7f, 0xf1, 0x72,
    0x20, 0xa8, 0x59, 0xcb, 0x3f, 0x29, 0xd4, 0x5a,
    0x33, 0x23, 0x75, 0xbc, 0xa6, 0x09, 0xa2, 0xca,
    0x5d, 0xe2, 0x73, 0x0b, 0x83, 0x1d, 0x0a, 0xe8,
    0xfd, 0x67, 0xc4, 0x43, 0x28, 0x9d, 0xc5, 0xc6,
    0x11, 0x83, 0x7d, 0xe0, 0xdf, 0x90, 0xb4, 0xeb,
    0xc4, 0xc2, 0xcd, 0xc3, 0x52, 0x1f, 0xc3, 0xc2,
    0xc4, 0xcc, 0x14, 0xc0, 0xa6, 0xbf, 0xbc, 0x95,
    0xc2, 0xda, 0xfa, 0xc1, 0x3a, 0xb2, 0xa0, 0x5d,
    0x5c, 0x55, 0x62, 0xf4, 0x82, 0x33, 0xd5, 0x2b,
    0x3b, 0x25, 0x1d, 0xfa, 0xbc, 0x70, 0xbb, 0x35,
    0x64, 0x1f, 0x70, 0x5d, 0x44, 0x90, 0x89, 0xec,
    0xe1, 0xed, 0x5a, 0xa2, 0x13, 0xa8, 0x6a, 0x0c,
    0xd7, 0xc1, 0x9e, 0x0f, 0x69, 0x97, 0xd8, 0x84,
    0x5d, 0xa9, 0xbd, 0x5d, 0x00, 0x39, 0x8c, 0x6a,
    0x9b, 0x3d, 0x57, 0xc1, 0x04, 0x7f, 0xe0, 0x0c,
    0xac, 0x4a, 0xd7, 0x0b, 0xfa, 0x90, 0x6c, 0xbd,
    0xb6, 0xd2, 0x03, 0x8d, 0xf0, 0x53, 0x23, 0xa8,
    0x78, 0xd7, 0x0d, 0xd4, 0xbd, 0x3d, 0xc6, 0x0e,
    0x74, 0xcf, 0x95, 0x9c, 0x9a, 0x74, 0xb6, 0x9f,
    0x3f, 0xfd, 0x61, 0x46, 0xed, 0x35, 0x3f, 0xec,
    0x36, 0x64, 0x6e, 0xed, 0x05, 0x36, 0xd1, 0xd7,
    0xf8, 0x64, 0x8e, 0xb9, 0x0a, 0xa0, 0xf0, 0xff,
    0x39, 0x38, 0x3f, 0xbb, 0x8d, 0xb1, 0x1c, 0xeb,
    0xe9, 0xcb, 0xa0, 0x24, 0x10, 0xb8, 0x51, 0xfc,
    0xf4, 0xca, 0x0a, 0x92, 0xca, 0xa3, 0xd4, 0xc6,
    0x76, 0x9b, 0x10, 0x8a, 0x45, 0x9e, 0x5f, 0x9a,
    0xe9, 0x35, 0x0c, 0x4a, 0x44, 0xa6, 0x97, 0x4e,
    0x87, 0x61, 0x5c, 0xee, 0xfb, 0x37, 0xf7, 0xd8,
    0x53, 0x3a, 0x13, 0x2f, 0xff, 0x07, 0x1d, 0x9c,
    0x70, 0x36, 0x4e, 0x8e, 0x35, 0x57, 0x9d, 0x57,
    0x77, 0x2d, 0x38, 0x97, 0x47, 0x32, 0x44, 0xe5,
    0xed, 0x16, 0xb0, 0x6c, 0x99, 0x5c, 0x66, 0x52,
    0x74, 0x04, 0x94, 0x3c, 0x11, 0xdb, 0x3c, 0xd4,
    0xc8, 0xc7, 0x45, 0x6c, 0xff, 0x75, 0x36, 0xef,
    0xc2, 0xe8, 0xf9, 0x37, 0x5d, 0x11, 0x4e, 0x19,
    0x5d, 0xbc, 0x09, 0xb6, 0xd4, 0xe0, 0x37, 0xb5,
    0x45, 0x82, 0x8c, 0x75, 0x32, 0x16, 0x9d, 0x9e,
    0x0f, 0x46, 0xdb, 0x09, 0x51, 0x08, 0x19, 0xdb,
```

```
    0xad, 0xa2, 0x5d, 0x2c, 0x3e, 0xa8, 0x05, 0x4a,
    0xff, 0xef, 0x2f, 0xfb, 0x9b, 0xf2, 0x7d, 0x7e,
    0x2d, 0x02, 0x49, 0x77, 0x2c, 0x92, 0xc3, 0x64,
    0xa2, 0xda, 0xbe, 0x24, 0x92, 0xef, 0x79, 0xc0,
    0xb1, 0xf4, 0x2f, 0x3d, 0x0a, 0x65, 0x1a, 0xac,
    0xf4, 0x1c, 0x5a, 0xf2, 0x4b, 0x98, 0xd6, 0x14,
    0x04, 0xec, 0xfe, 0x0d, 0xfc, 0xad, 0xbe, 0x3b,
    0xc9, 0x44, 0x9d, 0xc2, 0xe5, 0x3d, 0x6f, 0xb9,
    0x46, 0x1e, 0x21, 0xed, 0xea, 0x6c, 0xce, 0xbe,
    0xe8, 0xc1, 0x4f, 0xc3, 0x32, 0xc7, 0x26, 0xc5,
    0x5e, 0xe4, 0x2b, 0xf2, 0x72, 0x79, 0xf5, 0x19,
    0x72, 0x88, 0xbb, 0xbe, 0x93, 0x1a, 0x48, 0xd5,
    0x8a, 0x27, 0x49, 0x82, 0x2c, 0xe0, 0x98, 0xe0,
    0x73, 0x7f, 0xe5, 0xb9, 0xba, 0x7d, 0x55, 0x5f,
    0xf0, 0x66, 0xf9, 0x76, 0xcb, 0xca, 0xa2, 0xaa,
    0xf2, 0xc8, 0xaa, 0xab, 0xd6, 0x4d, 0x7c, 0xfd,
    0xa7, 0x1a, 0x7a, 0xb2, 0x9d, 0x6c, 0xa8, 0x49,
    0x8f, 0x21, 0xe9, 0x12, 0x88, 0x15, 0x50, 0x91,
    0x78, 0x6e, 0x9c, 0x1a, 0x50, 0xa4, 0x63, 0x22,
    0xa7, 0x7e, 0x76, 0x63, 0x7c, 0x32, 0xd0, 0x78,
    0xfb, 0x23, 0xae, 0xa9, 0x03, 0x1b, 0x34, 0x54,
    0xad, 0xb6, 0x2b, 0x96, 0x5a, 0x4a, 0xd9, 0x33,
    0xb6, 0x60, 0x55, 0xc7, 0xf6, 0x32, 0xba, 0xcc,
    0x8a, 0x7a, 0xf8, 0xef, 0x8c, 0xce, 0x57, 0x94,
    0x70, 0x71, 0x8a, 0xff, 0x20, 0x78, 0x1d, 0x4d,
    0x35, 0x7d, 0x7d, 0x42, 0xc6, 0x55, 0xcb, 0x6e,
    0x19, 0x3a, 0x68, 0x8a, 0x01, 0x87, 0xac, 0x19,
    0xa2, 0x93, 0x44, 0xcc, 0xc4, 0x0f, 0xf9, 0x13,
    0x00, 0x5a, 0xa5, 0x09, 0x06, 0x8f, 0x34, 0x9e,
    0x6a, 0xdf, 0x47, 0x35, 0x5d, 0xdd, 0xd6, 0xa7,
    0x42, 0x35, 0x90, 0x32, 0x33, 0xc9, 0x09, 0x73,
    0x40, 0x1b, 0x13, 0x20, 0x48, 0xa4, 0x36, 0xdc,
    0xe9, 0xb5, 0x9a, 0x53, 0xae, 0xdc, 0x15, 0xcb,
    0x70, 0xac, 0xd7, 0x8f, 0x0a, 0x05, 0xd4, 0xe2,
    0xc2, 0x7c, 0xc4, 0xa4, 0x40, 0xf9, 0x6a, 0x65,
    0x4a, 0x66, 0x3c, 0x9e, 0xc8, 0xb6, 0x7b, 0x4d,
    0xeb, 0x6c, 0xb8, 0xe3, 0x16, 0x94, 0x5c, 0x25,
    0xa2, 0xe9, 0x44, 0xce, 0xb9, 0x44, 0x2b, 0xee,
    0x51, 0x76, 0xcd, 0x2c, 0x1a, 0xeb, 0x86, 0xbe,
    0x3e, 0xc4, 0x9d, 0xdd, 0xd6, 0xb3, 0x11, 0xeb,
    0x1a, 0x38, 0x61, 0xb2, 0x4f, 0x9e, 0xcd, 0x91,
    0xab, 0x91, 0xb7, 0x98, 0xaf, 0xa9, 0x9c, 0x01,
    0x98, 0x65, 0x5b, 0xfc, 0x0a, 0xab, 0xce, 0x27,
    0xab, 0xa2, 0xff, 0x06, 0xd7, 0xac, 0xec, 0x01,
    0xe3, 0x32, 0x8c, 0x97, 0x13, 0x38, 0x3d, 0xa0,
    0x52, 0x37, 0x7f, 0x3c, 0x48, 0x91, 0x08, 0x6a,
    0x64, 0x87, 0x4d, 0xd2, 0x3d, 0x7a, 0x6e, 0xe5,
    0x85, 0x43, 0xf8, 0x57, 0xaf, 0xd7, 0x66, 0x9a,
    0xc4, 0x88, 0xe4, 0xcd, 0x37, 0xd2, 0xf4, 0x11,
    0x10, 0xda, 0xc0, 0x31, 0x0f, 0xd3, 0x09, 0x1c,
    0xe5, 0x74, 0x38, 0x95, 0x8a, 0x4a, 0x69, 0x3c,
    0xcf, 0x75, 0xd9, 0xac, 0x07, 0x68, 0xfa, 0x43,
    0xd8, 0x74, 0xa3, 0x12, 0xcd, 0x75, 0xdb, 0x1f,
    0xc4, 0xe8, 0xe1, 0x42, 0x3b, 0x64, 0x7a, 0x92,
    0x1c, 0x9b, 0xc4, 0xe4, 0x75, 0xd6, 0x0f, 0x07,
    0x3b, 0x37, 0x1f, 0x27, 0xf4, 0x18, 0x6c, 0x7c,
    0xb7, 0x4f, 0x23, 0xc8, 0xc4, 0x6d, 0x20, 0x36,
```

```
    0x56, 0x83, 0xf8, 0xa8, 0xc3, 0x52, 0x5a, 0xe8,
    0xec, 0xb7, 0x20, 0xc1, 0xcb, 0xa8, 0xaa, 0x1c,
    0x3a, 0x7a, 0x77, 0xae, 0x2c, 0xff, 0x71, 0x0c,
    0x9e, 0x41, 0xa9, 0xab, 0xf6, 0xc8, 0xf6, 0x71,
    0xcd, 0x1b, 0x7c, 0xc3, 0xcb, 0x1f, 0x98, 0xa7,
    0x72, 0xc2, 0x45, 0x8a, 0xc8, 0x91, 0x7b, 0xc3,
    0xf3, 0x62, 0xf2, 0xb7, 0x34, 0xd5, 0x0c, 0xca,
    0xd3, 0xaa, 0x8a, 0x63, 0xde, 0xa2, 0xaf, 0x99,
    0x82, 0x92, 0xb1, 0x88, 0x72, 0xf1, 0xe8, 0xda,
    0xe6, 0xca, 0x9e, 0xf9, 0xc9, 0xce, 0x2d, 0x66,
    0x2b, 0x72, 0x3b, 0x17, 0x94, 0x68, 0xba, 0xb3,
    0x22, 0x7c, 0x77, 0xce, 0xa0, 0x1b, 0x06, 0x45,
    0x19, 0x5c, 0xc2, 0x2e, 0x33, 0xca, 0xdc, 0x30,
    0x46, 0x21, 0xa0, 0x31, 0x4e, 0x7f, 0x73, 0xdf,
    0xf4, 0xff, 0x16, 0x72, 0x96, 0xb2, 0x21, 0x61,
    0x49, 0x97, 0x56, 0xd6, 0x83, 0xe9, 0x01, 0x0c,
    0xdc, 0x06, 0xd7, 0xd9, 0xcd, 0xce, 0x72, 0xbc,
    0x3e, 0x93, 0xd0, 0xba, 0x93, 0xe4, 0xcb, 0x8f,
    0x32,
  },
  /* bytes in plaintext */
  2065,
  /* associated data */
  {
    0x27, 0x47, 0xb0, 0x58, 0x61, 0x6c, 0x36, 0x10,
    0x5c, 0x77, 0x67, 0xe5, 0x7d, 0x1b, 0x31, 0x9a,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x2c, 0x51, 0xc9, 0x48, 0xd5, 0x1a, 0x83, 0x46,
    0xd2, 0x69, 0xb8, 0x2c, 0xc4, 0x47, 0x0a, 0x24,
    0x90, 0x49, 0x3a, 0x7e, 0xa1, 0xbd, 0x27, 0xa1,
    0x89, 0x8e, 0x1f, 0x8c, 0xeb, 0x49, 0xaa, 0xb9,
    0x83, 0x74, 0x73, 0x2a, 0x7b, 0x25, 0x62, 0x05,
    0x24, 0xaa, 0x63, 0x1b, 0x06, 0x40, 0x90, 0x4a,
    0x96, 0xa5, 0xdd, 0x40, 0x7f, 0xf5, 0xcb, 0x98,
    0x92, 0x91, 0xd5, 0x2c, 0x45, 0xbd, 0xe2, 0x7e,
    0xdc, 0xdb, 0x8b, 0xf3, 0xd2, 0x52, 0x76, 0xc0,
    0xf2, 0xa1, 0x6e, 0x2c, 0x83, 0x07, 0xbe, 0xff,
    0x51, 0x05, 0x1c, 0xe0, 0xd6, 0xab, 0x91, 0xc9,
    0xfd, 0xb3, 0x25, 0x82, 0xbb, 0x89, 0xf9, 0xef,
    0x30, 0x3f, 0x50, 0xdf, 0x36, 0xea, 0x1f, 0x2e,
    0x0a, 0xba, 0x48, 0x3e, 0xad, 0x88, 0x1e, 0xb9,
    0x3b, 0xd2, 0xf4, 0x66, 0x9f, 0x61, 0x0c, 0xd5,
    0xbe, 0x55, 0x4c, 0x71, 0x35, 0xd6, 0x36, 0xad,
    0xff, 0x87, 0xbf, 0xf8, 0x46, 0x0e, 0x66, 0xf7,
    0xe7, 0x98, 0xc1, 0xde, 0x67, 0x64, 0x66, 0x40,
    0xa8, 0x68, 0x4d, 0xd5, 0x0f, 0x3f, 0xd2, 0x6f,
    0xb8, 0xa4, 0x4e, 0x55, 0xc0, 0xae, 0x4a, 0x7a,
    0x9e, 0x47, 0x97, 0xb6, 0x43, 0x81, 0x17, 0x0e,
    0x57, 0x9e, 0xf6, 0x76, 0x80, 0x13, 0x61, 0xf7,
    0x03, 0x60, 0x01, 0x0b, 0xbc, 0x8d, 0xb9, 0x1b,
    0xb1, 0x6f, 0x4a, 0x9d, 0x4e, 0x77, 0xd3, 0x62,
    0xb9, 0x6c, 0xd8, 0x10, 0x20, 0x0f, 0x7e, 0xd6,
    0x9a, 0xa4, 0xf8, 0x34, 0xbd, 0x1c, 0x82, 0x1e,
```

```
    0xd7, 0x65, 0xe6, 0xf0, 0x4f, 0x3f, 0x4f, 0x78,
    0x39, 0x4f, 0x16, 0xdd, 0xad, 0x5e, 0x1e, 0xde,
    0x62, 0x3c, 0x1d, 0x1d, 0xa0, 0x87, 0x8d, 0xd7,
    0xbd, 0xa7, 0x8b, 0xf9, 0xa0, 0xd9, 0x53, 0xb6,
    0x8a, 0x34, 0x56, 0xa6, 0x35, 0x38, 0xb4, 0xdc,
    0xcd, 0xb6, 0x4e, 0xb9, 0xfc, 0x35, 0x8d, 0x4a,
    0x3f, 0x82, 0x34, 0x20, 0x38, 0xf4, 0x6d, 0x9d,
    0x55, 0x76, 0x19, 0xe7, 0x94, 0xbd, 0x38, 0x0a,
    0xed, 0x3b, 0xaa, 0x21, 0xff, 0x3d, 0x30, 0x57,
    0x6f, 0xd2, 0xcc, 0xf8, 0x91, 0x89, 0xe3, 0x4e,
    0x65, 0x51, 0x49, 0x8c, 0x66, 0x8e, 0x2c, 0xda,
    0x89, 0xe7, 0xb6, 0x23, 0xb1, 0x14, 0x31, 0xae,
    0x59, 0xa4, 0x5c, 0x5d, 0x86, 0xb3, 0xfd, 0x7f,
    0x1d, 0xe5, 0xb1, 0x85, 0x4e, 0xf9, 0x24, 0x0f,
    0xa2, 0x51, 0xd5, 0xb1, 0x0e, 0xc6, 0xfe, 0x06,
    0xe5, 0xf5, 0x8d, 0x8a, 0x04, 0xd3, 0x3c, 0xbc,
    0x55, 0xd6, 0xc7, 0xb8, 0x1f, 0x10, 0x2c, 0x98,
    0x04, 0xb3, 0x4c, 0x97, 0x91, 0x60, 0x8a, 0x53,
    0x45, 0x29, 0xe2, 0xcd, 0x05, 0xf6, 0x6e, 0x00,
    0x18, 0x9d, 0xf4, 0x13, 0x2c, 0xd3, 0xc0, 0x26,
    0x23, 0xeb, 0xef, 0x4a, 0x19, 0xb9, 0x31, 0x5b,
    0x57, 0x6b, 0x29, 0xc2, 0xaa, 0x10, 0xb7, 0xc7,
    0x31, 0x61, 0xb1, 0x80, 0x5b, 0xa2, 0xf0, 0x61,
    0xcd, 0x0c, 0xec, 0x9f, 0x1b, 0xde, 0x60, 0x79,
    0x9c, 0x00, 0xe6, 0x9d, 0x3c, 0x7c, 0x3f, 0x78,
    0x2c, 0x81, 0x05, 0x19, 0x22, 0x5c, 0x5b, 0x05,
    0x4d, 0x29, 0x4f, 0x61, 0x24, 0xb4, 0x57, 0xae,
    0xf2, 0xc7, 0x63, 0xe2, 0x80, 0xac, 0x3d, 0x6a,
    0xaa, 0x57, 0xac, 0x36, 0x3a, 0x62, 0x86, 0xf1,
    0x84, 0xee, 0xb9, 0x92, 0xd4, 0xa7, 0x97, 0xd5,
    0x97, 0xe4, 0x64, 0xc8, 0x33, 0xa8, 0x62, 0x47,
    0x52, 0xf5, 0x7d, 0xc6, 0x32, 0x20, 0x93, 0xb0,
    0x60, 0xe4, 0xd5, 0xea, 0xac, 0xcd, 0xe9, 0xe4,
    0x39, 0xb1, 0x9c, 0x44, 0x2c, 0xea, 0xf4, 0x8c,
    0xea, 0xd0, 0x3b, 0xe8, 0x7d, 0x57, 0x24, 0xe3,
    0xac, 0xf5, 0x22, 0x17, 0xcf, 0xfe, 0xfc, 0xdb,
    0x5b, 0xe4, 0x50, 0x11, 0xab, 0x15, 0x94, 0x01,
    0x7e, 0xd6, 0x2a, 0x1c, 0x87, 0x40, 0x68, 0x0d,
    0xd2, 0x93, 0x61, 0x3f, 0x4a, 0xe6, 0x36, 0xaf,
    0x87, 0x1b, 0x89, 0xc6, 0x2a, 0xc3, 0x86, 0xd1,
    0x7f, 0xbd, 0x70, 0x1e, 0x32, 0x7d, 0xc0, 0xd8,
    0xa0, 0xb3, 0x6b, 0x64, 0xf9, 0x5c, 0x5e, 0xee,
    0x39, 0x68, 0xc5, 0xc9, 0x07, 0x44, 0x39, 0x9b,
    0x6a, 0xbf, 0xd6, 0x29, 0x42, 0xb1, 0x22, 0x63,
    0x37, 0xed, 0x2f, 0xe7, 0x5a, 0xd1, 0xbe, 0x55,
    0x80, 0x13, 0x32, 0xf6, 0x91, 0x99, 0x53, 0x73,
    0x74, 0xe6, 0x3e, 0x40, 0xd3, 0x4a, 0x28, 0xd1,
    0x92, 0xec, 0xc6, 0x0a, 0x8b, 0xd4, 0x14, 0x4b,
    0x8f, 0xd4, 0x53, 0xe1, 0xfa, 0x3c, 0x69, 0x57,
    0xdb, 0x69, 0x83, 0xa9, 0x55, 0xde, 0x5b, 0xd9,
    0xae, 0x94, 0x99, 0x59, 0x8e, 0xe7, 0xa4, 0x66,
    0x3c, 0x86, 0x8a, 0xe1, 0x60, 0x03, 0xa8, 0xcc,
    0x9b, 0x31, 0x95, 0x10, 0xe7, 0x92, 0x32, 0xbc,
    0x3b, 0xe0, 0xc0, 0x57, 0x18, 0x32, 0x34, 0xd4,
    0x26, 0xa8, 0x13, 0x4b, 0x13, 0xdb, 0xe9, 0x18,
    0xde, 0xdb, 0xaf, 0xbc, 0x8c, 0x31, 0xf2, 0x24,
    0x1f, 0xa0, 0x66, 0x72, 0xd4, 0xf7, 0xc5, 0xdb,
```

```
    0xd4, 0x45, 0xf7, 0x84, 0x75, 0x5d, 0x7c, 0xcd,
    0x58, 0x7a, 0x72, 0x50, 0x6b, 0x0c, 0x9b, 0x6b,
    0xca, 0xaa, 0xa8, 0x50, 0xe7, 0x4f, 0x43, 0x45,
    0x4a, 0xd0, 0x92, 0x10, 0x5b, 0x7e, 0x10, 0xc3,
    0xbc, 0x90, 0x68, 0x8d, 0xc9, 0x2b, 0x36, 0xeb,
    0x29, 0x33, 0x39, 0xf5, 0x5b, 0x66, 0x25, 0x4d,
    0xe5, 0x65, 0x4c, 0xeb, 0x3a, 0x98, 0x70, 0x62,
    0x01, 0x27, 0xc9, 0x17, 0x5a, 0x68, 0xc1, 0xec,
    0x65, 0x02, 0xb8, 0x66, 0x20, 0xa5, 0x57, 0x26,
    0xfc, 0x8c, 0xf0, 0x75, 0x9c, 0x08, 0xc1, 0x02,
    0x20, 0x71, 0x19, 0x0f, 0xaf, 0xf0, 0x30, 0x48,
    0x8a, 0x32, 0xd1, 0xb1, 0x6d, 0xa5, 0xbf, 0x36,
    0x1f, 0x06, 0x32, 0xc8, 0x65, 0x61, 0xac, 0x6a,
    0x4f, 0x0d, 0x55, 0xb0, 0x71, 0xb5, 0x62, 0xf6,
    0x83, 0x8a, 0x65, 0xb4, 0xc6, 0x64, 0x0c, 0xa7,
    0x32, 0xc5, 0x1c, 0xf7, 0x23, 0x91, 0xe8, 0x1c,
    0x92, 0x14, 0xaf, 0x2f, 0x4d, 0x1a, 0xcb, 0x07,
    0x37, 0x95, 0xf8, 0x1c, 0xab, 0x0f, 0x6c, 0x66,
    0xe6, 0xd0, 0xd6, 0xd2, 0x48, 0x87, 0x68, 0x79,
    0xd3, 0x6e, 0x3a, 0x3f, 0x1f, 0xbe, 0x49, 0xd1,
    0xc2, 0x56, 0x26, 0x58, 0x5f, 0xd3, 0x36, 0xc1,
    0x99, 0x48, 0x86, 0x9c, 0xcb, 0xb7, 0x65, 0xf7,
    0x50, 0xa5, 0xd6, 0x29, 0x2c, 0xac, 0xb1, 0x95,
    0xcc, 0x28, 0x9a, 0x7b, 0x86, 0x2d, 0x31, 0x37,
    0xbd, 0xae, 0xa9, 0x4d, 0x3a, 0xd0, 0xcf, 0x86,
    0x4d, 0x7f, 0xc1, 0xb6, 0x81, 0x7a, 0xb4, 0xd5,
    0x9f, 0x16, 0xbe, 0x8d, 0x94, 0x82, 0xe5, 0x3d,
    0x73, 0x90, 0xf8, 0xc1, 0xa1, 0x77, 0x47, 0x73,
    0xae, 0xe2, 0x5e, 0x10, 0x89, 0xc0, 0x40, 0x02,
    0xe8, 0x9d, 0x38, 0xda, 0xeb, 0xd5, 0x87, 0x22,
    0x22, 0x10, 0xce, 0x5e, 0xab, 0xec, 0xba, 0xa8,
    0x05, 0xf1, 0x14, 0x25, 0x38, 0x88, 0x85, 0x01,
    0x96, 0xc8, 0x0c, 0x2b, 0x16, 0xb4, 0xf8, 0x2d,
    0x65, 0x2d, 0x2f, 0x00, 0xc8, 0x8f, 0x4f, 0x81,
    0xab, 0x56, 0xe9, 0x91, 0x6f, 0x92, 0x57, 0x27,
    0x16, 0x43, 0xc8, 0xf8, 0xf4, 0x3f, 0x45, 0x9e,
    0x11, 0x30, 0x77, 0xa5, 0x23, 0xae, 0x75, 0xf4,
    0x8d, 0x1a, 0x22, 0x37, 0x2e, 0xe9, 0xe8, 0x6e,
    0x5c, 0x80, 0xd3, 0x11, 0x92, 0x9a, 0x41, 0x64,
    0xc7, 0x7b, 0xc9, 0x46, 0x44, 0x97, 0x7e, 0x16,
    0xa7, 0x34, 0xdd, 0xe7, 0x24, 0x6d, 0x5c, 0x63,
    0x85, 0xf1, 0xe8, 0xc1, 0x4d, 0x08, 0xa7, 0xf2,
    0xe0, 0xf9, 0x42, 0x12, 0x62, 0x59, 0xc8, 0xc6,
    0x98, 0xd8, 0x0e, 0xf2, 0x77, 0x9d, 0x7d, 0x77,
    0x86, 0x08, 0x64, 0xab, 0xf1, 0xc6, 0x3f, 0x1c,
    0x3f, 0x4d, 0xec, 0x0b, 0x97, 0x6d, 0x38, 0x74,
    0xf8, 0x9f, 0x2b, 0x44, 0xe0, 0x73, 0x62, 0xc4,
    0xfa, 0x15, 0x1e, 0x56, 0x91, 0xe3, 0x37, 0xa5,
    0x03, 0xe0, 0x67, 0xe8, 0x97, 0x7d, 0xb1, 0xaf,
    0x96, 0xc9, 0xc1, 0xa9, 0x7d, 0x33, 0x75, 0x87,
    0x63, 0x80, 0xc3, 0x11, 0x4f, 0x9b, 0x14, 0x72,
    0x84, 0xbb, 0x6d, 0xc6, 0xba, 0xf8, 0x84, 0x2b,
    0x9b, 0x9b, 0xe3, 0x4d, 0xc9, 0x7f, 0xb0, 0xb6,
    0x8f, 0x4f, 0x01, 0x7d, 0xdc, 0x55, 0x2a, 0x54,
    0x6d, 0x22, 0x85, 0xed, 0x5b, 0x20, 0xa8, 0x66,
    0xef, 0x23, 0xea, 0xa5, 0x8a, 0x8d, 0x40, 0x93,
    0x45, 0xe9, 0x31, 0x11, 0x57, 0xed, 0x73, 0xa6,
```

```
    0x1c, 0x00, 0xb7, 0x48, 0xdf, 0xa3, 0x3d, 0x6f,
    0x4e, 0x3b, 0x10, 0xe0, 0xdc, 0xcd, 0x95, 0x76,
    0x37, 0x0c, 0x7d, 0x5e, 0xf7, 0xd3, 0x29, 0x8d,
    0xb6, 0x38, 0x52, 0x6f, 0x77, 0x17, 0xa7, 0xd3,
    0xf7, 0xf4, 0xa9, 0xea, 0x42, 0xb9, 0x93, 0x69,
    0x16, 0xc6, 0x51, 0x8c, 0x9f, 0x3b, 0x83, 0x6f,
    0xf7, 0xe9, 0xf4, 0x61, 0xce, 0xb9, 0xa8, 0x4d,
    0x65, 0x30, 0x97, 0xec, 0xbe, 0xb9, 0xc9, 0x3c,
    0xa0, 0xc9, 0xd0, 0x3f, 0xbb, 0xf9, 0x5c, 0x49,
    0x1a, 0x80, 0x7f, 0x92, 0xb8, 0x60, 0x64, 0x6b,
    0xa5, 0xb1, 0x28, 0x17, 0x30, 0xae, 0x87, 0x37,
    0xcd, 0x24, 0x75, 0x57, 0xb8, 0x3d, 0x17, 0x19,
    0x87, 0x44, 0xe0, 0x78, 0xb2, 0x7e, 0xfb, 0x50,
    0xb0, 0xed, 0xd7, 0xa1, 0x8a, 0x27, 0x59, 0x8d,
    0x65, 0x81, 0x62, 0x32, 0xf1, 0x5a, 0x65, 0xe5,
    0x52, 0xd7, 0xd9, 0xc6, 0x3a, 0x37, 0x2b, 0xd9,
    0x86, 0xf4, 0x57, 0x68, 0x77, 0xb4, 0x22, 0x04,
    0x71, 0x23, 0x20, 0x8e, 0x00, 0x78, 0xc9, 0x6c,
    0x54, 0x44, 0x30, 0x6a, 0xa7, 0x43, 0xbc, 0x1f,
    0xdf, 0xae, 0x2f, 0x2e, 0xf7, 0x15, 0x7b, 0x0a,
    0x7f, 0xfb, 0xc2, 0x48, 0x86, 0xbd, 0x29, 0xe7,
    0xeb, 0x66, 0x16, 0x6d, 0x7c, 0xb7, 0x68, 0x64,
    0x7c, 0xb3, 0x01, 0x4d, 0x67, 0x22, 0x6d, 0x43,
    0x96, 0x88, 0xbb, 0x3c, 0x11, 0xdd, 0x88, 0x07,
    0x2b, 0x77, 0x6c, 0xcd, 0xbc, 0x6a, 0x8b, 0xb6,
    0x7e, 0x40, 0x02, 0xa5, 0xd2, 0x23, 0xb4, 0x30,
    0x96, 0x8a, 0xaa, 0x20, 0xec, 0xd1, 0xe7, 0x79,
    0x3d, 0xb4, 0x83, 0xb6, 0xf4, 0x1a, 0xf7, 0x58,
    0x4b, 0x10, 0x29, 0x5d, 0x7f, 0xad, 0x28, 0xb7,
    0x91, 0xa2, 0xb9, 0x4c, 0xc7, 0xf7, 0x40, 0x3c,
    0x93, 0xbb, 0x83, 0x33, 0x72, 0x18, 0x8c, 0xda,
    0x25, 0x7e, 0xa0, 0x61, 0x76, 0xb2, 0x19, 0xab,
    0x9a, 0xfe, 0xce, 0x16, 0x99, 0xe3, 0x3e, 0x68,
    0x92, 0xba, 0x7c, 0xb3, 0xb0, 0x35, 0xa2, 0xd8,
    0xa1, 0xb5, 0xfc, 0x69, 0x1f, 0x30, 0xb2, 0xd0,
    0x23, 0xc1, 0x73, 0xf5, 0x24, 0x79, 0xbf, 0xa1,
    0xbb, 0xd3, 0x10, 0xaa, 0x85, 0x09, 0x2d, 0x77,
    0x3e, 0x9a, 0x63, 0x09, 0x12, 0x0b, 0x84, 0x27,
    0x7a, 0xb1, 0x58, 0x54, 0xad, 0x08, 0x73, 0x20,
    0x9d, 0x01, 0x78, 0x65, 0x07, 0x38, 0x22, 0x9c,
    0xb4, 0xdd, 0xec, 0x5b, 0x4b, 0x20, 0x6f, 0xef,
    0x69, 0x72, 0x89, 0xf1, 0x6d, 0xa9, 0xa3, 0x4d,
    0x8c, 0xfb, 0x7f, 0x31, 0x02, 0x7b, 0xff, 0x08,
    0x2c, 0xa7, 0xba, 0x3f, 0x9f, 0x4b, 0x2d, 0xa0,
    0x23, 0x84, 0x8f, 0x27, 0x57, 0x11, 0x9d, 0xcf,
    0xb2, 0xa7, 0xe4, 0x21, 0x18, 0xda, 0xa8, 0x02,
    0x87, 0x16, 0x73, 0x5e, 0xea, 0x3d, 0x0d, 0x7e,
    0xd0, 0x47, 0x45, 0x3e, 0xd5, 0xc7, 0x3f, 0xf0,
    0x11, 0x09, 0xef, 0xfb, 0x17, 0x7a, 0x55, 0xf6,
    0xbf, 0x6c, 0xe7, 0x2a, 0x4f, 0x23, 0xc5, 0x0d,
    0x75, 0x44, 0xab, 0x3b, 0x5a, 0x07, 0xd8, 0x70,
    0x3b, 0xfe, 0x4d, 0x9c, 0xbc, 0xd3, 0x8f, 0xf2,
    0x57, 0xd0, 0xba, 0x76, 0x7e, 0xd0, 0x15, 0x7a,
    0x50, 0x4c, 0x21, 0x53, 0xf3, 0x3c, 0x7e, 0x79,
    0xda, 0xc0, 0x2a, 0x1d, 0x14, 0x9c, 0xcc, 0xf3,
    0x16, 0xa5, 0x7f, 0xb6, 0x4b, 0x15, 0xa7, 0x14,
    0x78, 0xf2, 0x19, 0x42, 0xd4, 0x19, 0xce, 0x42,
```

```
    0x3a, 0xc6, 0x19, 0xcc, 0xd1, 0x62, 0x61, 0x3b,
    0x30, 0x4d, 0xba, 0xa4, 0xb8, 0xc1, 0xfe, 0xa2,
    0x9a, 0x2a, 0x0a, 0x75, 0xb8, 0x65, 0xf8, 0x5a,
    0xde, 0xe6, 0xeb, 0x78, 0x11, 0x0d, 0x53, 0x7d,
    0x2e, 0xc4, 0x33, 0xa6, 0x42, 0x18, 0x08, 0x32,
    0xcb, 0xb3, 0x0f, 0x17, 0x2f, 0xd4, 0x21, 0x6f,
    0xd7, 0xc6, 0xa9, 0x25, 0x45, 0xe5, 0x6c, 0x1c,
    0x79, 0x6e, 0xdd, 0x1b, 0x42, 0x43, 0x53, 0xf8,
    0x3b, 0x9b, 0x7a, 0xab, 0x29, 0x5d, 0x3e, 0x33,
    0x6a, 0x51, 0xfb, 0x73, 0x3d, 0xd5, 0xfa, 0x61,
    0xde, 0xe4, 0x52, 0xe5, 0xf6, 0xa6, 0x3b, 0x91,
    0xef, 0x89, 0x59, 0xe4, 0x48, 0xeb, 0x67, 0x0a,
    0xfe, 0x69, 0x15, 0x2a, 0x52, 0x76, 0x6e, 0x6f,
    0xea, 0xa3, 0x1b, 0x10, 0x7e, 0x19, 0xd7, 0x8f,
    0x75, 0xa9, 0x74, 0xe8, 0xa7, 0xb1, 0x0a, 0xff,
    0x1a, 0xbc, 0x43, 0xeb, 0x0d, 0x78, 0x3f, 0x80,
    0xa4, 0x5d, 0x7d, 0xae, 0xb3, 0x85, 0xf5, 0xb7,
    0x80, 0xd6, 0x54, 0x9b, 0x64, 0xa8, 0xc7, 0xde,
    0x37, 0x8e, 0xf3, 0x36, 0x4c, 0x02, 0x91, 0xe9,
    0xca, 0x6b, 0xf1, 0xa5, 0x3a, 0xaf, 0xbe, 0x8f,
    0xee, 0x68, 0xd9, 0x8a, 0x09, 0x88, 0xfc, 0xde,
    0x9b, 0x05, 0x2c, 0xa8, 0x84, 0xf7, 0xdd, 0xae,
    0xac, 0x94, 0x8a, 0x00, 0x99, 0x06, 0xd4, 0x68,
    0x9b, 0x6e, 0x97, 0x03, 0xa1, 0x09, 0x73, 0x2d,
    0x89, 0xf3, 0x6f, 0x9c, 0xb9, 0x6b, 0x4c, 0xae,
    0x58, 0x86, 0xb2, 0x0d, 0xc8, 0x8b, 0xa5, 0xf2,
    0x9a, 0xf7, 0x08, 0x16, 0x88, 0xfa, 0x52, 0xa7,
    0x8c, 0x35, 0xbb, 0x6d, 0x44, 0x07, 0xb4, 0x38,
    0x90, 0x4a, 0xd7, 0x08, 0x31, 0xcc, 0x7e, 0x2f,
    0xef, 0x52, 0x18, 0x67, 0xd8, 0xfc, 0x4b, 0x55,
    0x6d, 0x8c, 0x30, 0x7f, 0xc4, 0xa9, 0x2b, 0xf5,
    0x0b, 0x9a, 0xb1, 0xfc, 0x5e, 0x9f, 0x36, 0xcf,
    0xf0, 0x56, 0xbd, 0x46, 0x4c, 0xef, 0xfa, 0x09,
    0xaa, 0xb6, 0x29, 0xb3, 0x1d, 0x44, 0x97, 0x14,
    0xfa, 0xb7, 0x08, 0xf3, 0xc5, 0xd4, 0x48, 0x86,
    0xf7, 0x10, 0xe0, 0x57, 0x6a, 0x9b, 0xd1, 0x71,
    0x76, 0x86, 0xf7, 0x71, 0x71, 0x62, 0x07, 0x5a,
    0x08, 0x34, 0x64, 0x4e, 0x6a, 0x0a, 0x20, 0x97,
    0x29, 0xf4, 0xa3, 0xed, 0xe3, 0x53, 0xae, 0x4e,
    0xc6, 0x77, 0x6a, 0xd6, 0x97, 0x61, 0xe4, 0x51,
    0x4f, 0xb5, 0x59, 0x3c, 0x08, 0x51, 0xb8, 0xdb,
    0x34, 0xcd, 0xd3, 0x0e, 0xc0, 0xf4, 0xcd, 0x74,
    0xf6, 0xd5, 0x77, 0xb1, 0xf5, 0x23, 0x9c, 0x6a,
    0xf8, 0xf7, 0xbe, 0x68, 0x3d, 0x56, 0x87, 0x13,
    0x0d, 0x0f, 0x72, 0x1b, 0x49, 0xe5, 0x98, 0xbd,
    0x92, 0x7c, 0x87, 0xc3, 0xe4, 0x7e, 0x5b, 0xfc,
    0x14, 0x7d, 0x7c, 0x3c, 0xa3, 0x65, 0xd8, 0x21,
    0x4e, 0xaa, 0x84, 0xe0, 0xbc, 0x2a, 0xa6, 0x35,
    0xf0, 0x8c, 0x10, 0xef, 0x08, 0x05, 0x8a, 0xa1,
    0x8b, 0xfc, 0x43, 0xf3, 0xb9, 0x75, 0x5d, 0x3b,
    0x28, 0xa5, 0x1d, 0x18, 0x00, 0x23, 0xe0, 0xc5,
    0x07, 0x16, 0x2c, 0xaf, 0xab, 0x54, 0x10, 0xa0,
    0xbd, 0x62, 0xe3, 0xf2, 0x58, 0x12, 0x82, 0x8a,
    0x78, 0xb9, 0xa6, 0xb6, 0xc4, 0xec, 0xf3, 0xd1,
    0x87, 0x08, 0xb6, 0x33, 0x79, 0x4d, 0x00, 0x0e,
    0x70, 0x19, 0xf4, 0xfd, 0x1b, 0xe0, 0x0b, 0x17,
    0x10, 0x98, 0x67, 0xc6, 0xa7, 0x00, 0xd7, 0x75,
```

```
    0x29, 0xfd, 0xf4, 0xb2, 0x5c, 0x06, 0x40, 0xdb,
    0xc6, 0x7c, 0x35, 0xdd, 0x86, 0xd4, 0xb3, 0x79,
    0x51, 0x86, 0x4d, 0xe6, 0xfe, 0xb3, 0x9e, 0x16,
    0x81, 0x7b, 0x34, 0x96, 0x43, 0x2b, 0x24, 0xfe,
    0xf6,
  },
  &case9
};

test_case_t case11 = {
  /* key */
  {
    0xd6, 0xc2, 0xb8, 0x55, 0x32, 0xb0, 0x2e, 0xec,
    0x5a, 0x0e, 0x7e, 0xfc, 0x1a, 0x31, 0xdb, 0xef,
    0x2f, 0x4e, 0xcf, 0x01, 0xeb, 0x2a, 0xfd, 0x5d,
    0xed, 0x4e, 0x71, 0x77, 0xc0, 0xd4, 0x1e, 0x07,
    0x45, 0x20, 0x78, 0x6f, 0x13, 0xef, 0x99, 0xb1,
    0xe6, 0x2a, 0x98, 0xd2, 0x5b, 0x6e, 0xc5, 0x16,
  },
  /* bytes in key */
  48,
  /* plaintext */
  {
    0x18, 0x7d, 0xdd, 0x99, 0x8c, 0x75, 0xb5, 0x50,
    0xf8, 0x32, 0x94, 0x62, 0x77, 0x2a, 0xc0, 0xd6,
    0x36, 0x7b, 0x9c, 0x49, 0xdd, 0xf2, 0xfa, 0xde,
    0x0b, 0x02, 0x5e, 0x1c, 0x95, 0xbe, 0xb3, 0x7a,
    0x5b, 0x1e, 0x6c, 0x42, 0x8d, 0x45, 0xf1, 0x9d,
    0xde, 0xe0, 0x63, 0xd2, 0xf5, 0xe9, 0x56, 0x33,
    0x9d, 0x15, 0x0e, 0x07, 0x7a, 0xfd, 0xe4, 0xb8,
    0x29, 0x53, 0x32, 0x1c, 0x7f, 0x18, 0x0c, 0x2b,
    0xfb, 0x37, 0xa0, 0x28, 0xd3, 0x4a, 0x2e, 0x2f,
    0xc2, 0x11, 0xb7, 0xad, 0x62, 0x88, 0x6a, 0x6a,
    0x78, 0xb8, 0x7a, 0xf1, 0x53, 0xa3, 0x7f, 0xc1,
    0x96, 0xa6, 0xc4, 0xaa, 0xfc, 0x05, 0xee, 0xc2,
    0xad, 0xe7, 0x50, 0x1c, 0x41, 0x37, 0x1f, 0x79,
    0xa2, 0x2b, 0x3c, 0xda, 0x7c, 0x25, 0xb1, 0x4c,
    0xe0, 0x3a, 0xdd, 0x54, 0x23, 0x4f, 0xd8, 0xb2,
    0xe1, 0xba, 0x9c, 0xc5, 0x74, 0xc4, 0xc0, 0x46,
    0xd7, 0x7a, 0x05, 0x9f, 0x0a, 0x57, 0x97, 0xe4,
    0x18, 0x21, 0xb6, 0x48, 0x53, 0x23, 0x16, 0x9f,
    0xbf, 0x42, 0xe1, 0x27, 0xed, 0x7b, 0x85, 0x27,
    0x2d, 0xbc, 0x00, 0x1f, 0x96, 0x6d, 0xbb, 0xa8,
    0x9a, 0x0e, 0xf1, 0x6a, 0x22, 0x1f, 0xd8, 0x2f,
    0x37, 0x81, 0x64, 0x0b, 0xb3, 0xf4, 0x35, 0x76,
    0xc8, 0xe3, 0xf8, 0x33, 0xfb, 0x7f, 0xfa, 0xa9,
    0x0d, 0xa7, 0xfa, 0xc8, 0x8d, 0xc9, 0xc0, 0x7e,
    0xdf, 0x92, 0x59, 0xd0, 0xc7, 0xcb, 0xa7, 0x6d,
    0x50, 0x54, 0x18, 0xe3, 0xee, 0x2b, 0xf5, 0xa8,
    0xd4, 0x16, 0x10, 0xf7, 0xd4, 0xf8, 0xe8, 0x3f,
    0xd1, 0x7b, 0x95, 0x3b, 0x21, 0x98, 0x46, 0xba,
    0x88, 0x89, 0x2c, 0x68, 0x1e, 0x80, 0xfe, 0xca,
    0x90, 0x9d, 0x75, 0x33, 0x0e, 0xb5, 0x76, 0x01,
    0xf8, 0x7c, 0x1d, 0xac, 0xe7, 0x5e, 0x49, 0x46,
    0x20, 0x64, 0x1a, 0x53, 0xe0, 0xaa, 0xbd, 0xc9,
    0x3c, 0x72, 0x60, 0x36, 0xc8, 0xaa, 0xc0, 0xaa,
    0xb3, 0x5d, 0x4b, 0x75, 0x62, 0x6b, 0xef, 0x4e,
```

```
    0x78, 0xde, 0x0c, 0xb1, 0x3e, 0xbc, 0x1d, 0x03,
    0x83, 0x68, 0x35, 0x69, 0xad, 0x81, 0x9b, 0x79,
    0xf8, 0xa4, 0x7f, 0xfe, 0xe6, 0xe8, 0xdc, 0x58,
    0x66, 0x0e, 0xb1, 0xb1, 0x7e, 0xbc, 0x0f, 0xd9,
    0x03, 0xed, 0x3c, 0x05, 0x2f, 0x40, 0xc9, 0xec,
    0xe1, 0xc6, 0x90, 0x2a, 0x89, 0x0a, 0xd8, 0x78,
    0xb2, 0x01, 0x46, 0x02, 0x3e, 0x46, 0x5d, 0x2b,
    0xe9, 0x76, 0x10, 0xff, 0x13, 0xd7, 0xbd, 0x68,
    0x4b, 0xcb, 0x87, 0x51, 0x90, 0xb5, 0x4d, 0x7c,
    0x4f, 0x60, 0x75, 0x4f, 0x1a, 0x2d, 0xfd, 0x3a,
    0xd7, 0x10, 0xa4, 0x8b, 0x9f, 0x1d, 0x9b, 0x63,
    0x02, 0xe7, 0xc4, 0x37, 0x14, 0xc6, 0x1c, 0xa6,
    0xd6, 0xe6, 0x42, 0xaa, 0xd6, 0x1f, 0xe4, 0x00,
    0x29, 0xfe, 0x81, 0xb7, 0x13, 0xc3, 0x53, 0x03,
    0x6e, 0x9f, 0xfc, 0x77, 0x8e, 0x69, 0x61, 0x03,
    0x35, 0xee, 0x83, 0xbb, 0xa4, 0x55, 0x5b, 0xe9,
    0xe8, 0x0f, 0x15, 0xed, 0xeb, 0x5b, 0x43, 0xc5,
    0x31, 0x41, 0xd4, 0xa5, 0x9c, 0x9e, 0x6c, 0x92,
    0xce, 0x12, 0x3e, 0x3f, 0xd3, 0x1d, 0x5d, 0xc3,
    0xd7, 0x34, 0xa9, 0x06, 0xe1, 0x1c, 0xe9, 0x1f,
    0x64, 0x98, 0x33, 0x62, 0x92, 0x86, 0x0b, 0xd4,
    0x4e, 0x5f, 0x18, 0xe3, 0x6d, 0xee, 0xb7, 0xaf,
    0x7d, 0x8e, 0xeb, 0x27, 0x54, 0xc5, 0xf7, 0xd1,
    0x34, 0x67, 0x0b, 0xd1, 0xae, 0xa0, 0x12, 0xd2,
    0x05, 0x9a, 0xbb, 0xe1, 0x97, 0x36, 0x23, 0x4d,
    0x61, 0x0d, 0xe2, 0x0b, 0x94, 0x38, 0xd8, 0x73,
    0x34, 0x25, 0x7c, 0x64, 0xdc, 0x07, 0xb6, 0x21,
    0xc4, 0xe8, 0x88, 0xcf, 0x19, 0xc5, 0x4f, 0x0e,
    0xb4, 0x14, 0xaf, 0xad, 0x0f, 0x1f, 0xf2, 0x40,
    0x93, 0x5b, 0xe0, 0x98, 0x96, 0x3e, 0x2f, 0xda,
    0x3f, 0x80, 0x57, 0x55, 0x3a, 0xb8, 0x2e, 0x2e,
    0xe0, 0x60, 0xdd, 0xe5, 0x5b, 0xcc, 0xef, 0x12,
    0xf7, 0x91, 0x2b, 0x24, 0xd7, 0xb6, 0x73, 0x88,
    0x1e, 0x7e, 0xc7, 0x84, 0x15, 0xe5, 0xf4, 0xbe,
    0x78, 0x66, 0xf6, 0xe3, 0x9f, 0x2d, 0x2d, 0xc7,
    0x81, 0x28, 0xe1, 0x62, 0x57, 0x80, 0x2f, 0x9f,
    0xc9, 0x69, 0x01, 0x66, 0xc5, 0xb0, 0x48, 0x05,
    0x18, 0x43, 0x1d, 0xa9, 0x8c, 0x9d, 0x75, 0x74,
    0x81, 0x67, 0x9a, 0x39, 0xfd, 0xa6, 0xff, 0x86,
    0xce, 0xdb, 0x60, 0x32, 0xe9, 0xf2, 0x41, 0x82,
    0x04, 0xc6, 0x83, 0xc7, 0xe1, 0x4b, 0xfa, 0x61,
    0xa3, 0x5d, 0x09, 0xc8, 0xee, 0x27, 0xc6, 0xcd,
    0x16, 0xb4, 0x19, 0xc6, 0x13, 0x69, 0x90, 0x90,
    0x2c, 0x93, 0x87, 0x51, 0xd9, 0xcd, 0x68, 0xce,
    0x0a, 0x87, 0x35, 0xe0, 0x5e, 0x47, 0x0b, 0x1f,
    0xa1, 0x15, 0x52, 0x11, 0xb4, 0x36, 0xf2, 0xa4,
    0xe0, 0xe5, 0x50, 0x1b, 0x81, 0xce, 0xa9, 0xd6,
    0x5e, 0x71, 0xb3, 0x60, 0x79, 0x0e, 0x7e, 0xd1,
    0x06, 0x24, 0xf9, 0xcb, 0xb9, 0x4f, 0xe4, 0x78,
    0x17, 0x97, 0xaf, 0xa6, 0x5e, 0xee, 0xd8, 0x7c,
    0x09, 0x25, 0xc9, 0xe6, 0xe0, 0xbb, 0x7f, 0x70,
    0xdf, 0x43, 0x65, 0x6e, 0x7a, 0xfb, 0x4d, 0x26,
    0xd0, 0x05, 0xf9, 0x2d, 0x60, 0x86, 0x71, 0x34,
    0x27, 0xce, 0xff, 0x68, 0xba, 0x68, 0xfa, 0xb6,
    0x8a, 0x5c, 0x65, 0x0e, 0x7e, 0xc5, 0x70, 0x26,
    0x48, 0x9f, 0x5a, 0xc5, 0x65, 0x9b, 0xab, 0xb7,
    0x49, 0x70, 0x71, 0xfe, 0x9c, 0x54, 0x7f, 0x17,
```

```
    0x1d, 0xe3, 0x70, 0xe3, 0x60, 0x8d, 0x46, 0xba,
    0xfd, 0xc9, 0xf2, 0xf1, 0xa4, 0x96, 0xe6, 0x5d,
    0x4a, 0x62, 0x46, 0xdf, 0x3c, 0x8d, 0x6e, 0x6c,
    0xeb, 0x81, 0x3c, 0x98, 0x11, 0xee, 0x18, 0xaf,
    0x4c, 0x84, 0x15, 0xc6, 0x9f, 0x3d, 0xa7, 0x21,
    0x53, 0x78, 0xac, 0xf6, 0x47, 0xab, 0x38, 0x8a,
    0xc3, 0x5e, 0xde, 0x5a, 0xe9, 0x6d, 0x85, 0xff,
    0xbc, 0x55, 0x86, 0xd8, 0x90, 0x9f, 0x1f, 0x73,
    0xca, 0x71, 0x3c, 0xfb, 0x41, 0x58, 0x70, 0xfe,
    0xee, 0x72, 0xb2, 0x78, 0x55, 0x5d, 0xfc, 0x5b,
    0xed, 0x87, 0xf1, 0x97, 0x0c, 0xda, 0xea, 0x2b,
    0x94, 0x9b, 0xea, 0x1c, 0x22, 0xfe, 0x66, 0x6e,
    0xe8, 0xd1, 0x77, 0x5a, 0x11, 0xe5, 0xb0, 0xe9,
    0xaf, 0x29, 0x3e, 0xa0, 0xb5, 0xa4, 0xc9, 0xf5,
    0x0a, 0xa6, 0x48, 0xdf, 0x64, 0xaf, 0xb4, 0x45,
    0x13, 0x3d, 0x96, 0x49, 0xf6, 0xfb, 0x79, 0xa3,
    0x90, 0x5f, 0xd7, 0x02, 0xcf, 0x81, 0x64, 0x24,
    0xf2, 0x64, 0x8f, 0x73, 0x31, 0xe6, 0x9a, 0xf2,
    0x8e, 0xf4, 0xae, 0xb6, 0x6e, 0x6e, 0x25, 0x1f,
    0x01, 0xc9, 0x31, 0x3b, 0x7e, 0xb5, 0xb4, 0xf7,
    0x6a, 0x8f, 0xa7, 0x38, 0xf1, 0x37, 0x3c, 0x14,
    0x0d, 0xaf, 0xf4, 0x71, 0x5b, 0xb4, 0x29, 0xf2,
    0xe9, 0xcd, 0x1f, 0x72, 0x9a, 0xed, 0x8a, 0xf6,
    0xa9, 0xab, 0x09, 0x27, 0x5e, 0x43, 0xeb, 0xee,
    0xae, 0xf0, 0x5f, 0x93, 0xb2, 0xd8, 0x06, 0x73,
    0xb9, 0x7f, 0x08, 0x71, 0xeb, 0x53, 0xa1, 0x03,
    0x72, 0x46, 0x3a, 0x10, 0x85, 0xc1, 0xa3, 0x05,
    0x67, 0x08, 0x4b, 0x4b, 0x58, 0xfb, 0x8e, 0x53,
    0x91, 0x00, 0xfe, 0x66, 0xcd, 0x2c, 0x2a, 0xd0,
    0xbd, 0x0c, 0xd9, 0x7d, 0xf3, 0x3c, 0x10, 0xd9,
    0x4c, 0xe2, 0x3e, 0x51, 0x2c, 0xa7, 0x9b, 0x2b,
    0x2a, 0x22, 0xc7, 0x47, 0x96, 0x15, 0x53, 0x92,
    0x1e, 0x2e, 0x4a, 0x82, 0x7f, 0xd8, 0x0f, 0x93,
    0xb7, 0xb3, 0x4a, 0x2b, 0x7b, 0x7d, 0x9b, 0xb3,
    0xaa, 0x74, 0x55, 0x4a, 0xe2, 0x02, 0xbc, 0x37,
    0x3e, 0x02, 0xe8, 0x73, 0x34, 0x27, 0x42, 0xdb,
    0xd6, 0xf7, 0x28, 0xad, 0x63, 0x05, 0x84, 0x24,
    0x0d, 0x7b, 0x41, 0xce, 0xf7, 0x16, 0xd2, 0x52,
    0x77, 0x44, 0xb9, 0x77, 0xc9, 0x48, 0xab, 0x9d,
    0x55, 0x5c, 0xfa, 0x2b, 0xb7, 0x1d, 0xae, 0xba,
    0x0e, 0x7d, 0x72, 0x6a, 0xba, 0x3a, 0xbb, 0x08,
    0x2f, 0xba, 0xb5, 0xb5, 0xcb, 0x39, 0x88, 0x1d,
    0xa7, 0x90, 0x57, 0x6c, 0x20, 0xb3, 0x7b, 0xc6,
    0x50, 0x19, 0xee, 0xd6, 0xdf, 0x20, 0x46, 0x94,
    0x26, 0x3d, 0xa5, 0xf4, 0x9d, 0x10, 0xf8, 0x5b,
    0xcf, 0x77, 0x0f, 0x5e, 0x24, 0x0d, 0xb8, 0x40,
    0xd7, 0x25, 0x63, 0x5f, 0x2e, 0x1b, 0x99, 0x27,
    0xc0, 0x9a, 0xa7, 0xf9, 0x9a, 0xac, 0x1c, 0x01,
    0x0b, 0xa6, 0xb9, 0x75, 0xe7, 0xb8, 0x17, 0x26,
    0x71, 0x49, 0x82, 0x60, 0x38, 0x9f, 0x22, 0x47,
    0x83, 0x48, 0x70, 0x2d, 0x48, 0xf3, 0xb7, 0xdc,
    0xcc, 0x58, 0xee, 0x59, 0x28, 0xaf, 0xf4, 0xc4,
    0xe8, 0xbf, 0xa2, 0xf9, 0x02, 0x52, 0xc8, 0xf4,
    0xa5, 0x82, 0xbd, 0xf1, 0xd7, 0x56, 0x8f, 0x80,
    0xf6, 0x8a, 0x17, 0x5d, 0xff, 0xbe, 0xae, 0x2d,
    0x20, 0xe3, 0x87, 0x11, 0x02, 0x19, 0xd4, 0x65,
    0x15, 0x71, 0x09, 0x2d, 0x78, 0x99, 0x4a, 0x75,
```

```
    0x03, 0x83, 0x38, 0x8a, 0x88, 0x47, 0xdc, 0xd3,
    0xcf, 0xff, 0x2e, 0x96, 0x19, 0xf0, 0xf5, 0x01,
    0xe8, 0x9b, 0xde, 0x9c, 0xb4, 0xb7, 0xa0, 0x03,
    0xa0, 0x24, 0xb7, 0x74, 0x27, 0xa6, 0xc9, 0xa9,
    0xd3, 0x94, 0x40, 0x44, 0x1f, 0x2c, 0x36, 0xe0,
    0x6b, 0xa6, 0x27, 0x84, 0x18, 0xb5, 0xad, 0x7f,
    0x93, 0x28, 0x3c, 0x86, 0xce, 0x6d, 0x57, 0x36,
    0x72, 0x56, 0x1a, 0x09, 0x2e, 0x00, 0xcd, 0x4b,
    0x19, 0xf0, 0x09, 0x65, 0xa2, 0x21, 0xed, 0xbd,
    0x89, 0xe4, 0xc8, 0xa6, 0x05, 0xd8, 0xe9, 0xaa,
    0xff, 0xdb, 0xed, 0xfe, 0x4f, 0xa5, 0x67, 0x6d,
    0xbc, 0xef, 0x8b, 0xda, 0x28, 0x9b, 0x72, 0xda,
    0xe8, 0x7f, 0x9f, 0xd9, 0x59, 0xb7, 0x29, 0x90,
    0xc6, 0x37, 0x7a, 0xaf, 0xa0, 0x50, 0x18, 0xdd,
    0x76, 0xf2, 0x66, 0xc6, 0x2e, 0xff, 0x5c, 0x77,
    0xf9, 0x1a, 0x07, 0xb0, 0xce, 0x64, 0x5e, 0xf2,
    0x8a, 0xa2, 0xd3, 0xa8, 0xe0, 0xcc, 0x82, 0x69,
    0xbd, 0x01, 0xf7, 0xde, 0xc3, 0x43, 0x3e, 0xb1,
    0x2f, 0x7c, 0x6b, 0x18, 0x16, 0x05, 0xf2, 0xa2,
    0x06, 0x3d, 0x40, 0x73, 0x6c, 0x47, 0xf0, 0x6b,
    0xd6, 0x2d, 0x8a, 0x77, 0x84, 0x1a, 0x80, 0xd5,
    0x93, 0x7e, 0x7b, 0x9d, 0xec, 0x05, 0xa4, 0x03,
    0x47, 0xed, 0x4c, 0x89, 0x4b, 0xdd, 0xf7, 0x36,
    0x1f, 0xf8, 0x5a, 0x94, 0x22, 0xf2, 0xdd, 0x3f,
    0x8e, 0x49, 0x9c, 0xe7, 0x9e, 0xe6, 0xc2, 0x78,
    0x18, 0x30, 0x42, 0xba, 0x0b, 0x11, 0x5a, 0x72,
    0x55, 0x98, 0x1b, 0xf2, 0x16, 0x0a, 0x6e, 0x44,
    0x68, 0x8c, 0x69, 0x28, 0x2c, 0xd8, 0x8a, 0x0f,
    0x69, 0x1a, 0x7f, 0xeb, 0xe1, 0x26, 0x18, 0x70,
    0x51, 0x96, 0x21, 0x99, 0x72, 0xed, 0x19, 0x6a,
    0x0a, 0x2a, 0xc8, 0xe5, 0x15, 0xaa, 0xc3, 0x07,
    0xdb, 0x44, 0xe7, 0xe1, 0xc3, 0xc2, 0x00, 0x21,
    0xe0, 0xf0, 0x55, 0xe1, 0x66, 0x59, 0xe8, 0xfb,
    0x0d, 0xdb, 0x3f, 0xda, 0xb7, 0x04, 0xf6, 0x13,
    0x7a, 0x4b, 0x4d, 0x87, 0x42, 0x92, 0x7c, 0x63,
    0xac, 0xbb, 0x7f, 0x40, 0x2a, 0xc9, 0xce, 0xe2,
    0x92, 0x6b, 0xb4, 0xd2, 0xd1, 0x72, 0x2c, 0x99,
    0xf8, 0x3e, 0xc7, 0xec, 0xdc, 0xc8, 0xf8, 0x76,
    0x34, 0xb3, 0x26, 0x73, 0xaf, 0xb4, 0x00, 0x84,
    0xf2, 0x71, 0xc3, 0xea, 0xa9, 0x66, 0x99, 0xcd,
    0xd3, 0x5a, 0x6c, 0xdb, 0x55, 0x9c, 0x4a, 0x3c,
    0x67, 0xd0, 0x31, 0xd7, 0x00, 0x11, 0x4d, 0x9a,
    0x79, 0x52, 0xdf, 0x40, 0x8e, 0x5b, 0x01, 0x80,
    0x4a, 0xa9, 0xbe, 0x34, 0xa6, 0x0b, 0x7f, 0xf7,
    0x84, 0xaa, 0x24, 0x2e, 0x74, 0x17, 0x9d, 0x8c,
    0x95, 0x74, 0x43, 0xa8, 0xbf, 0xec, 0x17, 0x14,
    0x47, 0xa9, 0xe8, 0xaa, 0x49, 0x2d, 0x1e, 0xab,
    0xb5, 0xea, 0x44, 0x9a, 0xa2, 0x78, 0xfd, 0xb2,
    0x98, 0x32, 0x6c, 0xd2, 0x78, 0x82, 0xa0, 0x79,
    0x74, 0x85, 0x9f, 0x83, 0xe7, 0x0b, 0x23, 0xe5,
    0xbe, 0x85, 0xf8, 0x40, 0x4c, 0xde, 0x8e, 0xa0,
    0x2b, 0xaa, 0xbc, 0xe4, 0x7a, 0x73, 0xe9, 0x1e,
    0x5e, 0x59, 0x1e, 0xf8, 0x30, 0x5a, 0x2d, 0xe4,
    0x1c, 0x87, 0xa4, 0xc7, 0xa2, 0x81, 0x87, 0xc2,
    0x78, 0xbf, 0x2f, 0xa0, 0xad, 0x04, 0x44, 0x2b,
    0x0c, 0x9d, 0xc6, 0x2a, 0x5b, 0xf3, 0xac, 0xb8,
    0x8d, 0x3e, 0xe9, 0x89, 0x8c, 0xf8, 0x49, 0x7a,
```

```
    0x20, 0x09, 0x0d, 0x83, 0x22, 0xdf, 0x65, 0x85,
    0x63, 0x71, 0x25, 0xdf, 0x8f, 0x5a, 0xcc, 0x7a,
    0xcf, 0x65, 0xcb, 0xd0, 0x2a, 0xa6, 0xc4, 0xfc,
    0xad, 0x12, 0xf3, 0x61, 0x5b, 0x56, 0xb9, 0x3c,
    0x38, 0x96, 0xbb, 0xfd, 0x57, 0xda, 0x9f, 0x98,
    0xf1, 0x4d, 0xaa, 0x86, 0xd8, 0x52, 0xf1, 0x4a,
    0xed, 0xf9, 0xb3, 0x7b, 0x5c, 0xec, 0x7e, 0x4c,
    0xc6, 0xf8, 0xaf, 0xe1, 0xc7, 0xac, 0x04, 0xbe,
    0x12, 0x39, 0x43, 0xe5, 0x6a, 0x49, 0x63, 0xbc,
    0xe0, 0xdf, 0xc0, 0x01, 0xac, 0xf0, 0x1f, 0xb9,
    0xec, 0x27, 0x90, 0x81, 0xe5, 0xaf, 0xf0, 0xb8,
    0x9a, 0xdb, 0x03, 0x07, 0x6a, 0x1a, 0xf1, 0xb9,
    0x09, 0xb6, 0xc2, 0x2a, 0x6d, 0xc0, 0x28, 0x64,
    0xf6, 0x42, 0x7b, 0x53, 0x8b, 0x2d, 0x1f, 0x5b,
    0x8d, 0xa8, 0x48, 0xf4, 0x95, 0xa8, 0x35, 0x05,
    0x1e, 0xac, 0x3d, 0x92, 0x89, 0x39, 0x66, 0x14,
    0xe6, 0x3b, 0x81, 0xf8, 0xe8, 0x5b, 0xaf, 0xd5,
    0x80, 0xec, 0x20, 0x7f, 0x9a, 0x3d, 0xed, 0xf0,
    0x6a, 0x11, 0x84, 0x59, 0x73, 0xe6, 0x70, 0x8f,
    0xc2, 0x01, 0xf7, 0xcd, 0x77, 0x23, 0x50, 0x80,
    0xa2, 0xa7, 0x7d, 0x0e, 0x87, 0x4e, 0x1b, 0x67,
    0xff, 0x13, 0x6d, 0xc8, 0xee, 0x7e, 0x6e, 0x01,
    0x46, 0x4c, 0x82, 0x8a, 0x4f, 0x78, 0x3c, 0x00,
    0x8d, 0x17, 0x25, 0x6b, 0x43, 0x21, 0x9c, 0xd8,
    0x9d, 0x90, 0x81, 0xce, 0x95, 0x22, 0x27, 0xee,
    0x34, 0x4a, 0x09, 0x2d, 0x05, 0x7a, 0xdf, 0x5d,
    0xdf, 0x4b, 0x61, 0x8b, 0x2f, 0xc7, 0xf5, 0x85,
    0x7f, 0x83, 0x6b, 0x25, 0x36, 0x23, 0x22, 0xb7,
    0x37, 0x93, 0x43, 0xec, 0x21, 0x78, 0x71, 0xf6,
    0x6c, 0xab, 0x19, 0x7f, 0x21, 0xc4, 0x0f, 0x45,
    0x5f, 0xac, 0x49, 0x67, 0x51, 0x6d, 0x81, 0x4c,
    0x04, 0x96, 0xe3, 0x86, 0xa5, 0xdc, 0x91, 0xa8,
    0x8b, 0x3c, 0xb4, 0x3d, 0xfa, 0x6f, 0x68, 0x61,
    0xad, 0x2c, 0xbd, 0xbe, 0x20, 0x83, 0x6f, 0x3e,
    0xfc, 0x48, 0xb1, 0x11, 0xd9, 0xbf, 0x0e, 0xb8,
    0xa0, 0xa8, 0x56, 0x46, 0x69, 0xb2, 0xc5, 0xd6,
    0xf5, 0x1c, 0xf6, 0x86, 0x5d, 0xfe, 0x63, 0x80,
    0x2c, 0x44, 0x75, 0x8e, 0x87, 0x38, 0x98, 0x02,
    0x21, 0xc5, 0x7d, 0xee, 0x8a, 0xf9, 0x4b, 0x35,
    0x03, 0xad, 0x7c, 0x8c, 0xf5, 0x77, 0x2b, 0x4e,
    0x51, 0xeb, 0xd4, 0xc3, 0xfe, 0x3d, 0xab, 0x95,
    0xd0, 0x07, 0xb5, 0xde, 0xe0, 0x31, 0x29, 0xcb,
    0xf9, 0x2f, 0xd9, 0x4e, 0x7c, 0x26, 0x0d, 0xca,
    0x3b, 0x43, 0x92, 0xce, 0x69, 0x3f, 0x74, 0x82,
    0x0f, 0x49, 0xa0, 0x58, 0xb5, 0xee, 0x62, 0xb4,
    0x7b, 0x56, 0x57, 0x6b, 0xfa, 0x6b, 0xd7, 0x5d,
    0xa3, 0x39, 0x93, 0xc6, 0xa8, 0xb6, 0x9b, 0x87,
    0xb3, 0xd0, 0x11, 0x46, 0x92, 0xbe, 0x35, 0xec,
    0xcb, 0xae, 0xb5, 0x67, 0x03, 0x5f, 0x01, 0xe7,
    0x55, 0x8f, 0xa3, 0x46, 0x82, 0x2b, 0x36, 0x35,
    0x12, 0xe9, 0xa4, 0xcd, 0xcf, 0x15, 0x50, 0x3b,
    0x29, 0x91, 0xd8, 0x35, 0xff, 0xd5, 0xb2, 0xed,
    0x9b, 0x89, 0x5a, 0x8f, 0xbf, 0x98, 0x49, 0x2a,
    0xa7, 0x13, 0xc9, 0xd2, 0x05, 0x2f, 0x68, 0x5e,
    0xc9, 0xdc, 0x5d, 0xc0, 0xbd, 0xe5, 0x00, 0x07,
    0x38, 0x7a, 0x21, 0x7c, 0x27, 0x4f, 0x64, 0x5b,
    0x35, 0x1b, 0x99, 0x57, 0xf8, 0x7d, 0x8e, 0xc1,
```

```
    0xd0, 0x6f, 0xa5, 0xff, 0xb9, 0x8f, 0xa2, 0x57,
    0x6d, 0xf9, 0x7f, 0x28, 0x3f, 0x3a, 0x42, 0x84,
    0xed, 0xfd, 0x1e, 0x8d, 0xe9, 0x79, 0xa7, 0xfb,
    0x54, 0x3b, 0xbd, 0x47, 0xe2, 0xd0, 0xc5, 0x47,
    0x25, 0x39, 0x9c, 0x3e, 0x74, 0x23, 0xd4, 0xa8,
    0xdb, 0x1f, 0x1b, 0x64, 0x31, 0x55, 0x18, 0x22,
    0xf7, 0xa1, 0x5b, 0x9d, 0x0c, 0xb0, 0x19, 0x7d,
    0xa3, 0x18, 0x83, 0xc1, 0xdc, 0x6c, 0x91, 0xf0,
    0xcf, 0xd8, 0x34, 0xd2, 0xcc, 0x93, 0x50, 0x9a,
    0xc4, 0xdb, 0x7a, 0xb4, 0xb1, 0xd4, 0x38, 0xca,
    0x09, 0x2e, 0x4f, 0x19, 0x15, 0x38, 0xe4, 0xdd,
    0x93, 0x4d, 0x47, 0xe6, 0xc6, 0x54, 0x80, 0xba,
    0x94, 0x53, 0x14, 0x44, 0x45, 0x24, 0xdf, 0xbf,
    0x80, 0x8a, 0xb3, 0x5e, 0x58, 0x05, 0x22, 0x6e,
    0x1c, 0x84, 0xfa, 0x53, 0x7f, 0x64, 0xfd, 0x9c,
    0x68, 0x98, 0x5a, 0x9a, 0xf6, 0x27, 0x5f, 0x17,
    0xd6, 0x99, 0xc8, 0x4f, 0x39, 0x9b, 0x84, 0xa0,
    0x33, 0xde, 0x20, 0x0e, 0x46, 0xc8, 0x52, 0x30,
    0x39, 0xb5, 0x5e, 0x11, 0xb3, 0xd2, 0x16, 0x83,
    0x96, 0x65, 0x39, 0xaa, 0x76, 0xbc, 0x21, 0x21,
    0xaf, 0x13, 0x59, 0x7f, 0xab, 0xf9, 0xf1, 0xbd,
    0x7b, 0x89, 0xac, 0xa6, 0xaa, 0x21, 0x05, 0x69,
    0xcd, 0x1b, 0x42, 0x36, 0x69, 0xe8, 0x13, 0xae,
    0xb6, 0x1d, 0x23, 0xc4, 0x6c, 0x4d, 0x9e, 0xa4,
    0xfe, 0x61, 0x1e, 0x6d, 0x67, 0x11, 0x33, 0x94,
    0x7c, 0xc3, 0x2b, 0x10, 0xef, 0x26, 0x7b, 0xdf,
    0x28, 0x45, 0x5c, 0xde, 0xbd, 0xb4, 0xf5, 0x51,
    0x6e, 0xff, 0xe9, 0x13, 0xe0, 0xda, 0xd2, 0x48,
    0x87, 0x43, 0x14, 0x47, 0xed, 0xc6, 0xb4, 0xd4,
    0xb7, 0x76, 0xee, 0x97, 0xce, 0x47, 0x1f, 0x95,
    0xec, 0x05, 0x83, 0x2a, 0x0c, 0x37, 0x6b, 0x35,
    0xb9, 0xa7, 0xdf, 0xc6, 0xa4, 0x65, 0xc7, 0x1a,
    0x6a, 0x32, 0xf1, 0x8d, 0x99, 0xd9, 0xad, 0x14,
    0x0f, 0xb2, 0xa9, 0xc7, 0x4a, 0x00, 0xbe, 0x77,
    0x4c, 0x0c, 0x25, 0x7a, 0xaa, 0x1c, 0x06, 0x8e,
    0x70, 0x10, 0x75, 0xc6, 0x38, 0x0a, 0xf6, 0xa5,
    0x97, 0xb2, 0x52, 0x00, 0x44, 0xf9, 0xe5, 0x10,
    0xf3, 0x83, 0x23, 0x66, 0xe7, 0x56, 0xcc, 0x4d,
    0xfa, 0x99, 0x2b, 0xfc, 0xd1, 0x97, 0xc4, 0x89,
    0xfe, 0xd0, 0xe1, 0x1a, 0xfc, 0x8f, 0x7c, 0x45,
    0x82, 0x5a, 0x21, 0xe0, 0xc3, 0x61, 0x2d, 0xe7,
    0xf0, 0x9b, 0xcd, 0x94, 0x2b, 0x3c, 0x63, 0x87,
    0xb4, 0x77, 0x08, 0x6a, 0xf2, 0xf2, 0xe8, 0x44,
    0xd4, 0xa0, 0x19, 0x16, 0x43, 0xcf, 0x00, 0xbb,
    0x9a, 0x15, 0xc5, 0x11, 0x19, 0x26, 0xd0, 0xdf,
    0x93, 0x5f, 0x4a, 0x67, 0x18, 0x41, 0x5b, 0xde,
    0x71, 0x24, 0xf6, 0xd0, 0x02, 0x28, 0x3e, 0xa9,
    0xcc, 0x7b, 0xd5, 0xa7, 0x0b, 0x5a, 0x2b, 0x7c,
    0x3a, 0xac, 0x43, 0x93, 0xa5, 0x33, 0x7e, 0xd4,
    0xab, 0x57, 0x9d, 0x31, 0x06, 0xcd, 0x02, 0xfc,
    0x11, 0xf8, 0x6a, 0x50, 0xb3, 0x29, 0x1f, 0xd7,
    0x88, 0x1a, 0x4c, 0x62, 0xc7, 0x8a, 0x7f, 0xd8,
    0x74, 0xcb, 0x6c, 0xeb, 0xaa, 0x83, 0x2b, 0x76,
    0x47, 0x12, 0xb7, 0x96, 0xf7, 0x1a, 0x16, 0xb0,
    0x2c, 0x2a, 0x7a, 0x46, 0x32, 0x12, 0x1f, 0x17,
    0xf1, 0xde, 0x36, 0xff, 0xa5, 0x52, 0xd5, 0xa6,
    0x97, 0xda, 0x05, 0xd7, 0x60, 0x05, 0x41, 0x9f,
```

```
    0x1c, 0x0e, 0xd0, 0xcc, 0x82, 0xe7, 0x74, 0x6d,
    0x0e, 0x89, 0x61, 0x28, 0xfc, 0xa2, 0x9f, 0xb5,
    0xca, 0x17, 0x2d, 0xd7, 0x53, 0xb1, 0xe4, 0xd1,
    0xe9, 0xca, 0x26, 0x72, 0xca, 0x10, 0x25, 0xf5,
    0x27, 0x12, 0x84, 0xaf, 0x40, 0x9e, 0xb5, 0x2e,
    0x8d, 0xda, 0x8d, 0xfd, 0xb1, 0x3c, 0x9a, 0x2c,
    0xb7, 0xe5, 0x0b, 0xe0, 0x57, 0x9a, 0xf6, 0xd2,
    0x3a, 0x82, 0x47, 0xfa, 0xe9, 0x80, 0xb2, 0x4e,
    0x46, 0xe6, 0x71, 0x15, 0x17, 0xa4, 0x28, 0xb6,
    0x49, 0x96, 0xf4, 0x3f, 0x09, 0x65, 0xa7, 0x48,
    0xbc, 0x3b, 0xda, 0x00, 0xcf, 0xfc, 0x6f, 0x9e,
    0xe4, 0xbc, 0x72, 0x4e, 0x7e, 0xc1, 0x1b, 0x53,
    0xb4, 0xbd, 0x8a, 0x36, 0xf9, 0x87, 0x45, 0xe5,
    0x28, 0x52, 0xc2, 0x0d, 0xc7, 0x5d, 0xf0, 0x58,
    0x7d, 0xca, 0x65, 0x62, 0xe1, 0x38, 0x7d, 0x41,
    0x8f, 0x2f, 0x56, 0xda, 0xcd, 0xfd, 0x34, 0xfb,
    0x3c, 0x02, 0xb5, 0xd4, 0x3f, 0x8f, 0xd5, 0xa8,
    0x56, 0x85, 0x09, 0x70, 0x98, 0xbd, 0xcc, 0x96,
    0xed, 0xba, 0x06, 0x79, 0xde, 0xa4, 0xf5, 0x4f,
    0x5d, 0x83, 0x03, 0xcc, 0xee, 0x09, 0xf8, 0xce,
    0x08, 0x84, 0x03, 0x01, 0xbf, 0x14, 0x06, 0xde,
    0x7c, 0xce, 0x86, 0xa6, 0x25, 0xa1, 0x72, 0xe9,
    0x92, 0x18, 0xdb, 0x76, 0x06, 0x28, 0xaf, 0xdd,
    0x5a, 0x11, 0x37, 0x15, 0xda, 0xa0, 0x7b, 0x18,
    0x0c, 0x43, 0x91, 0x50, 0xc6, 0x9a, 0x79, 0x4d,
    0x4f, 0xd3, 0x11, 0xbe, 0x5d, 0xec, 0x87, 0xd2,
    0x64, 0x25, 0xa8, 0x0d, 0x06, 0x76, 0xf5, 0x21,
    0x18, 0x15, 0xe1, 0x93, 0x10, 0x0f, 0xd7, 0x9a,
    0x14, 0xf7, 0xe8, 0xc5, 0x16, 0x8d, 0x28, 0xdb,
    0x65, 0xe9, 0x47, 0x20, 0x47, 0xd7, 0x6f, 0x81,
    0x56, 0x73, 0x8b, 0xce, 0xaa, 0x6d, 0xb5, 0x3b,
    0x66, 0x66, 0xdd, 0x71, 0x67, 0x36, 0xc6, 0x4e,
    0x59, 0xc3, 0x26, 0x1a, 0x44, 0x23, 0xf9, 0x1c,
    0x42, 0xcc, 0xa7, 0xe0, 0xa2, 0xf5, 0x7b, 0x9a,
    0x57, 0x50, 0x06, 0x2e, 0x59, 0x78, 0x21, 0x8b,
    0x48, 0x98, 0x89, 0x5e, 0x7c, 0xd8, 0x0a, 0x52,
    0xe6, 0x75, 0xfb, 0x21, 0xd5, 0x97, 0xb1, 0x24,
    0xee, 0x33, 0x04, 0xf1, 0x93, 0x23, 0xc0, 0x35,
    0xb3, 0xdb, 0xba, 0xb5, 0xcd, 0x4d, 0x16, 0x0f,
    0xd6, 0x5f, 0x59, 0xca, 0x47, 0x3b, 0xc7, 0xe4,
    0xa0, 0x01, 0xe2, 0x0a, 0xd0, 0x1b, 0xde, 0x61,
    0x52, 0x4c, 0x70, 0x7f, 0x7c, 0x45, 0xbf, 0x5a,
    0x88, 0x23, 0x30, 0x3f, 0x64, 0x42, 0x10, 0x7a,
    0x36, 0x9e, 0xfd, 0xb6, 0x3c, 0x60, 0xb2, 0x0e,
    0x0a, 0xa3, 0x91, 0x85, 0x52, 0x4d, 0xfa, 0x76,
    0x32, 0xc0, 0x27, 0xf0, 0x85, 0xd7, 0xf5, 0xd8,
    0x19, 0xf6, 0x8f, 0xe4, 0x49, 0xe7, 0x0e, 0x10,
    0x7d, 0xfa, 0x3f, 0x45, 0xda, 0xe6, 0x3c, 0x40,
    0x6e, 0xea, 0x28, 0x86, 0x2a, 0x4c, 0xb0, 0x34,
    0x86, 0x83, 0x0d, 0x5e, 0x38, 0x9c, 0x78, 0xbe,
    0x77, 0xcc, 0x28, 0xcf, 0xfb, 0x8f, 0x6f, 0xc6,
    0x7e, 0x95, 0x11, 0x49, 0x63, 0x6f, 0x8a, 0xfb,
    0x8a, 0x63, 0x1b, 0xcf, 0x8a, 0x57, 0x3b, 0xfd,
    0x1a, 0x8e, 0xc2, 0xb4, 0x46, 0xb4, 0x84, 0xfe,
    0x9c, 0xd3, 0x27, 0x15, 0x70, 0x00, 0xf1, 0x75,
    0x67, 0xcc, 0x70, 0x4d, 0x44, 0xb0, 0xf1, 0x14,
    0x18, 0xc9, 0x05, 0xb8, 0x2a, 0x0b, 0x14, 0xd7,
```

```
    0x4d, 0x7a, 0x2c, 0x29, 0x47, 0x3a, 0x95, 0x67,
    0x1a, 0xcd, 0x9d, 0x19, 0x88, 0x61, 0xd6, 0x95,
    0xeb, 0xc0, 0x89, 0x6f, 0x20, 0xa2, 0xa6, 0x48,
    0xbf, 0x56, 0xdc, 0xbc, 0xd9, 0x73, 0x57, 0x2b,
    0x0d, 0xc0, 0x8b, 0xcf, 0xee, 0x25, 0x61, 0xf3,
    0x81, 0x6a, 0xf5, 0x1c, 0xf4, 0xee, 0x70, 0xd5,
    0xed, 0x29, 0x17, 0x79, 0x25, 0x2c, 0xf7, 0xc6,
    0xc6, 0xc6, 0xf3, 0x50, 0xf5, 0xf5, 0x02, 0x62,
    0xdc, 0xb7, 0x12, 0x99, 0x2a, 0xc8, 0x16, 0xe3,
    0x50, 0x57, 0x6d, 0xd3, 0xfb, 0x06, 0x24, 0xbf,
    0x9a, 0x2d, 0x66, 0x38, 0x1d, 0xd4, 0x03, 0x71,
    0xb1, 0x9b, 0xc2, 0xfb, 0x41, 0xbb, 0x7f, 0x9a,
    0xfd, 0x93, 0x74, 0xc3, 0xc3, 0xd9, 0x61, 0x49,
    0x3c, 0xe4, 0x0b, 0xaa, 0x85, 0x70, 0xc5, 0x36,
    0xd6, 0x51, 0x77, 0xe6, 0xbb, 0xe3, 0xf3, 0xe0,
    0x85, 0xda, 0x7b, 0x86, 0x45, 0xf9, 0x04, 0x4b,
    0xad, 0xe1, 0x6c, 0xef, 0xf8, 0x4b, 0xb4, 0x22,
    0x30, 0x0a, 0xd5, 0xd7, 0xb2, 0x3c, 0x01, 0xdf,
    0x8c, 0xc5, 0x14, 0x8f, 0x66, 0x99, 0x3b, 0x04,
    0x08, 0xdb, 0x0d, 0xb2, 0x58, 0x87, 0xd4, 0x7b,
    0xdb, 0x74, 0x93, 0x1a, 0x7f, 0x48, 0xc3, 0x65,
    0xcd, 0x6a, 0x76, 0xc7, 0x9a, 0x26, 0x53, 0xe4,
    0xa2, 0xe1, 0x2b, 0x0d, 0x7e, 0x4c, 0x81, 0x3f,
    0x93, 0x2d, 0xf6, 0xa8, 0xc9, 0x21, 0xb6, 0x2a,
    0x2e, 0x64, 0xaf, 0xa6, 0x9f, 0xc1, 0x99, 0x97,
    0x8d, 0x27, 0x47, 0xb0, 0x58, 0x61, 0x6c, 0x36,
    0x10, 0x5c, 0x77, 0x67, 0xe5, 0x7d, 0x1b, 0x31,
    0x9a, 0xc4, 0xdf, 0x50, 0x73, 0x99, 0x95, 0x30,
    0x24, 0x6d, 0x76, 0x2f, 0x25, 0xa6, 0xe8, 0x0b,
    0x37, 0x78, 0x9c, 0x1b, 0x10, 0x3f, 0xa2, 0x97,
    0x68, 0x55, 0x4e, 0x05, 0x32, 0x0b, 0xa6, 0x66,
    0x88, 0x42, 0x70, 0xc4, 0xda, 0x19, 0x4f, 0xf5,
    0x1a, 0xab, 0x80, 0x6b, 0x59, 0x9f, 0x54, 0x32,
    0xb1, 0x88, 0x39, 0x1f, 0x80, 0x3f, 0x24, 0xc8,
    0xd3, 0x1a, 0x2a, 0x0b, 0x51, 0x67, 0x35, 0xf5,
    0x3f, 0xdb, 0x53, 0xff, 0xdb, 0xfd, 0xaa, 0x57,
    0xdb, 0x1e, 0x82, 0xfb, 0x65, 0x7f, 0xd7, 0xd3,
    0x89, 0x25, 0xbc, 0x3a, 0x72, 0x98, 0x56, 0x9f,
    0x3f, 0x7d, 0xe2, 0x75, 0x76, 0x40, 0xa8, 0x04,
    0x88, 0xfa, 0x38, 0x6c, 0x30, 0xba, 0x60, 0x4c,
    0xec, 0xf1, 0x96, 0x6f, 0xe3, 0xa4, 0x60, 0x6a,
    0xbe, 0x3a, 0xec, 0x16, 0xb6, 0x76, 0xd6, 0xe9,
    0x80, 0x7e, 0x28, 0x21, 0xd7, 0xd2, 0x0d, 0xbc,
    0xa9, 0xb2, 0x73, 0x8f, 0x7b, 0x6c, 0x92, 0x91,
    0xa4, 0x06, 0xa1, 0xa0, 0x73, 0x8d, 0xdb, 0x8c,
    0x1c, 0xc9, 0x6b, 0x43, 0x00, 0x64, 0x43, 0xf5,
    0x27, 0x62, 0x5c, 0x24, 0xb3, 0x82, 0x7c, 0x7b,
    0xc4, 0x1f, 0x72, 0x10, 0x6a, 0x4a, 0xe4, 0xe3,
    0x99, 0x28, 0x69, 0x57, 0x91, 0x64, 0xa5, 0xa2,
    0xa4, 0x94, 0x38, 0x77, 0xbc, 0xc1, 0xf1, 0x82,
    0x00, 0x41, 0xec, 0x0d, 0x16, 0xde, 0xfc, 0xb1,
    0x01, 0xa0, 0xd5, 0x5c, 0x4c, 0x70, 0xf8, 0xf7,
    0xaf, 0xd3, 0x3c, 0x88, 0xad, 0x63, 0x87, 0xeb,
    0xf0, 0xf6, 0x13, 0xa5, 0xf5, 0x4f, 0xdc, 0x1e,
    0xc5, 0xc3, 0x2f, 0x13, 0x65, 0x0b, 0x3f, 0x19,
    0x83, 0x52, 0xeb, 0x33, 0xa5, 0xfd, 0x9e, 0x0a,
    0x14, 0x33, 0x92, 0xb1, 0x1c, 0x66, 0x2b, 0x3d,
```

```
    0x11, 0x66, 0x4f, 0xc7, 0x81, 0x9c, 0x05, 0x8e,
    0x05, 0x81, 0xa2, 0x5c, 0xd3, 0x26, 0x11, 0x26,
    0x40, 0xf4, 0x4e, 0xca, 0x96, 0xd3, 0x36, 0xec,
    0xf3, 0x90, 0xe1, 0x3a, 0xcc, 0x12, 0x23, 0x21,
    0x49, 0x72, 0xa7, 0x43, 0xcd, 0xb0, 0xf1, 0x74,
    0x2b, 0x6e, 0x73, 0x9f, 0x56, 0x1d, 0x0a, 0x15,
    0xad, 0xb5, 0xb1, 0xe8, 0xdf, 0xa1, 0xb1, 0xa9,
    0x06, 0xee, 0x4a, 0xe6, 0xac, 0x07, 0x99, 0x3e,
    0x74, 0x69, 0x78, 0xd3, 0x4a, 0x42, 0xd8, 0xd9,
    0x8c, 0x55, 0x35, 0xf0, 0x83, 0xa2, 0xa3, 0xac,
    0xf0, 0xcd, 0xb6, 0xd7, 0x7b, 0x4a, 0x35, 0x81,
    0xf5, 0x8b, 0x67, 0x9b, 0xc2, 0x03, 0xf4, 0xdc,
    0x07, 0xe0, 0xe6, 0xf2, 0x99, 0xa6, 0x7f, 0x4d,
    0xf4, 0xee, 0xbe, 0x17, 0x14, 0x34, 0xd0, 0xaa,
    0x7d, 0x14, 0x80, 0x27, 0x04, 0x9f, 0x93, 0x53,
    0xa6, 0x2b, 0xde, 0xb6, 0x6d, 0x10, 0x64, 0x92,
    0x66, 0x65, 0x44, 0x1d, 0xec, 0x46, 0xb4, 0xdd,
    0x2b, 0x78, 0x95, 0x84, 0x65, 0xb6, 0xcc, 0x26,
    0x36, 0x09, 0x8e, 0xf1, 0x6d, 0x9c, 0x56, 0xfd,
    0xf4, 0xff, 0x16, 0xa1, 0x95, 0x45, 0x55, 0x94,
    0x50, 0x4c, 0xe1, 0xb2, 0xe8, 0xad, 0x67, 0x0d,
    0xd9, 0x95, 0x76, 0xac, 0xfe, 0x0c, 0xb7, 0x04,
    0xe7, 0x67, 0x81, 0x33, 0xbf, 0xae, 0x50, 0x4e,
    0xb5, 0xef, 0xff, 0x43, 0x41, 0x27, 0xb7, 0x60,
    0xff, 0x33, 0x29, 0x37, 0x0d, 0x49, 0xc9, 0xd8,
    0x25, 0x73, 0xf9, 0x80, 0x3e, 0x1d, 0x86, 0x9c,
    0xed, 0xa6, 0x41, 0x77, 0x80, 0x9f, 0x44, 0x7e,
    0x7d, 0x07, 0x91, 0x33, 0x87, 0xa2, 0x5e, 0xa4,
    0x9c, 0xc1, 0x84, 0xba, 0x4a, 0x0a, 0x70, 0xf2,
    0x05, 0x9d, 0x5a, 0xea, 0x6e, 0xe7, 0xbd, 0x66,
    0x59, 0xe5, 0xc0, 0x09, 0x52, 0x97, 0x54, 0x9d,
    0x64, 0x7e, 0xb0, 0xf9, 0xd3, 0xc5, 0xd7, 0xe3,
    0x53, 0x09, 0x37, 0xaf, 0x72, 0x6d, 0x50, 0x64,
    0x03, 0x26, 0x28, 0x27, 0x64, 0x91, 0x43, 0x59,
    0x10, 0x2b, 0x61, 0x8c, 0xe4, 0x05, 0xc3, 0xf0,
    0xce, 0xb7, 0xf0, 0x4a, 0x93, 0x47, 0xb5, 0xcf,
    0x44, 0xf4, 0xee, 0xb9, 0xbe, 0x18, 0xfe, 0xe2,
    0xb9, 0xe0, 0x98, 0xa5, 0x30, 0x0d, 0x3c, 0x43,
    0xa5, 0xd2, 0x4b, 0xf0, 0x2b, 0x21, 0x8a, 0x25,
    0xd9, 0x50, 0xc0, 0x91, 0xc2, 0x5b, 0xd1, 0x20,
    0xac, 0xbe, 0xd3, 0xd1, 0x0c, 0x45, 0x38, 0x4e,
    0x8d, 0xe3, 0xfa, 0x60, 0xbf, 0xac, 0x96, 0x99,
    0x1f, 0x9d, 0x1e, 0x10, 0x11, 0x79, 0xb9, 0x75,
    0x24, 0xf9, 0x84, 0x48, 0x0c, 0x15, 0xb0, 0xaf,
    0x94, 0x06, 0x28, 0xdc, 0x72, 0x6a, 0xea, 0x8d,
    0xa4, 0x2a, 0x6e, 0xf9, 0x9d, 0x7e, 0x74, 0x37,
    0xa1, 0xd3, 0x75, 0x42, 0x2d, 0x44, 0x44, 0xc5,
    0xef, 0x7a, 0x4d, 0xe4, 0xbd, 0xe7, 0x51, 0x24,
    0xa2, 0x5f, 0x03, 0x21, 0xdc, 0x4c, 0xdc, 0xb4,
    0x98, 0x70, 0x35, 0xe3, 0x6d, 0xf2, 0xbc, 0x4e,
    0x87, 0x43, 0x2a, 0x60, 0x3c, 0x39, 0xae, 0x53,
    0x8d, 0xb0, 0xa6, 0x12, 0x54, 0x23, 0x80, 0x15,
    0x3b, 0x42, 0x2c, 0x93, 0x65, 0x45, 0x7a, 0xbd,
    0x1c, 0xdc, 0xf1, 0xb8, 0x5f, 0x86, 0xab, 0xcb,
    0x23, 0xe3, 0xf2, 0x1c, 0x50, 0x84, 0x19, 0x3e,
    0x5c, 0xc9, 0x2a, 0x5d, 0x36, 0x13, 0x96, 0xc1,
    0x5c, 0xc5, 0x1b, 0x1b, 0x9c, 0xaa, 0x6f, 0x93,
```

```
    0xe8, 0xd1, 0x16, 0x4e, 0x57, 0x39, 0x1f, 0x94,
    0x4b, 0x15, 0x22, 0xf4, 0xe1, 0xbd, 0xd2, 0xf0,
    0x58, 0xdc, 0xfa, 0x5d, 0x44, 0x1b, 0xa1, 0x25,
    0xc4, 0xc9, 0xa4, 0xf1, 0xeb, 0x29, 0x84, 0x8b,
    0x01, 0x2a, 0x15, 0x97, 0x3d, 0x23, 0x36, 0x00,
    0xd2, 0xb1, 0xee, 0xff, 0xf3, 0x06, 0x6d, 0xa8,
    0x22, 0xc3, 0x53, 0xa0, 0x21, 0x5f, 0x78, 0xc3,
    0x32, 0x7b, 0xba, 0x0f, 0x0e, 0xfb, 0xec, 0x15,
    0x9f, 0x67, 0x4b, 0x18, 0xf4, 0xac, 0xe5, 0x4d,
    0x9e, 0xbd, 0x8b, 0xd1, 0x35, 0xae, 0xbf, 0x5b,
    0xe8, 0x34, 0x15, 0x1b, 0xc6, 0x4d, 0x7d, 0x24,
    0x2d, 0x80, 0x42, 0x21, 0xa0, 0xf8, 0xb8, 0x06,
    0x50, 0x6e, 0x40, 0xed, 0xf7, 0x85, 0xcc, 0x3a,
    0x71, 0x8a, 0x5c, 0x42, 0x33, 0x75, 0xb8, 0x74,
    0x88, 0xdb, 0x2a, 0x91, 0x6d, 0x56, 0xa6, 0x33,
    0x72, 0x0d, 0xe3, 0xf5, 0x3e, 0x86, 0xff, 0xc5,
    0xcb, 0xe6, 0xb6, 0x13, 0x82, 0x70, 0xe6, 0x66,
    0xd1, 0xb1, 0x55, 0x5d, 0x6b, 0x3f, 0x35, 0xcd,
    0x3d, 0xf6, 0x0f, 0xaf, 0x78, 0xaa, 0x8a, 0xbd,
    0x3d, 0x8d, 0x7b, 0xa3, 0xe9, 0x19, 0x8e, 0x42,
    0x50, 0x2c, 0xa9, 0x64, 0x1c, 0x7e, 0x78, 0x48,
    0x1b, 0xa1, 0xfa, 0x3e, 0x51, 0x66, 0xca, 0x8e,
    0x47, 0xae, 0x9c, 0x65, 0x73, 0x45, 0x8e, 0xfa,
    0xa4, 0xcb, 0x33, 0x4f, 0x47, 0xb3, 0x24, 0x16,
    0x0f, 0x45, 0x2e, 0x89, 0x77, 0xcf, 0xae, 0x63,
    0x15, 0xcb, 0x24, 0xd6, 0x01, 0x59, 0x7a, 0x7d,
    0x40,
  },
  /* bytes in plaintext */
  4129,
  /* associated data */
  {
    0x8d, 0xcf, 0xe3, 0xb4, 0x53, 0xb9, 0x68, 0xc5,
    0x4f, 0x73, 0x62, 0xf2, 0xa4, 0x89, 0x19, 0x23,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x17, 0xee, 0x74, 0x8b, 0x55, 0xed, 0xc8, 0x90,
    0xa1, 0xf6, 0x4b, 0x12, 0xd0, 0x9e, 0xf8, 0xf7,
    0x85, 0x6c, 0xef, 0x00, 0xe8, 0x74, 0x99, 0x9b,
    0x3d, 0x01, 0x17, 0x62, 0x62, 0xeb, 0x0f, 0x17,
    0x93, 0x9f, 0x41, 0xdf, 0x8c, 0x2b, 0xe9, 0x3f,
    0xa5, 0xcb, 0x5a, 0xc1, 0x2e, 0x96, 0x41, 0x3a,
    0x03, 0xea, 0x4e, 0x85, 0xfe, 0x7f, 0x46, 0x42,
    0xd4, 0xd2, 0x74, 0xa6, 0xfd, 0x9f, 0x17, 0xc1,
    0x45, 0xd2, 0x5e, 0xc5, 0x5e, 0x03, 0xa2, 0x95,
    0xf7, 0x0f, 0xd7, 0x60, 0x6d, 0x26, 0xc3, 0x91,
    0x01, 0x37, 0xc5, 0x2e, 0xc2, 0xfe, 0xaf, 0x08,
    0x41, 0x3b, 0x7d, 0xb0, 0x11, 0x6c, 0x95, 0xcf,
    0xfe, 0xc7, 0xd4, 0x0a, 0x41, 0x1d, 0xe1, 0x38,
    0x9f, 0x09, 0xd9, 0x96, 0xba, 0xce, 0x50, 0x4e,
    0x94, 0x27, 0x5f, 0x04, 0xe0, 0x96, 0xea, 0xdf,
    0x91, 0x59, 0xc6, 0x74, 0xa5, 0x7a, 0x47, 0x57,
    0xbc, 0x98, 0x5b, 0xd4, 0xa4, 0x2f, 0x91, 0x77,
    0x59, 0x93, 0x50, 0x9f, 0xf0, 0x4d, 0x60, 0x90,
```

```
    0x99, 0x4a, 0xa7, 0xb8, 0xa8, 0x76, 0x7d, 0xed,
    0xf1, 0xc0, 0x7d, 0x9b, 0x37, 0x3e, 0x8b, 0x64,
    0x5b, 0x05, 0x1b, 0x26, 0x21, 0x4b, 0x49, 0x7d,
    0x46, 0x1c, 0xef, 0x4c, 0x2a, 0x1f, 0x62, 0xbe,
    0x0d, 0x0d, 0xae, 0xe7, 0x3d, 0x0d, 0x0f, 0x59,
    0x59, 0x18, 0xd8, 0x73, 0x96, 0x77, 0x93, 0x79,
    0x26, 0x25, 0xde, 0x5f, 0xbc, 0x24, 0x1d, 0xf7,
    0xd4, 0xeb, 0x72, 0x37, 0x7d, 0x91, 0xe9, 0x9d,
    0x8d, 0x5d, 0x0b, 0x15, 0xe3, 0x5f, 0xd4, 0xbf,
    0xb7, 0x8a, 0xe1, 0x35, 0x6d, 0x86, 0xa2, 0x6a,
    0x64, 0x30, 0xf6, 0x33, 0x47, 0xbe, 0xbe, 0x53,
    0x7d, 0x72, 0x9b, 0xbc, 0x30, 0x93, 0x40, 0xee,
    0xbd, 0xa9, 0xb0, 0xc5, 0x93, 0x63, 0x6e, 0x48,
    0x4c, 0xc3, 0x6e, 0x6a, 0xb8, 0x9c, 0xb4, 0xc0,
    0x3a, 0x08, 0xb1, 0x10, 0xfa, 0x18, 0xe9, 0x8c,
    0x68, 0x2d, 0xe3, 0x4d, 0xd6, 0x42, 0x87, 0x8e,
    0xa2, 0xb4, 0xdf, 0x9b, 0xf1, 0x64, 0x7a, 0x32,
    0xdf, 0xb2, 0x68, 0xf2, 0x9c, 0x70, 0xc2, 0x6a,
    0xb2, 0xa4, 0x07, 0xcc, 0x50, 0x57, 0x37, 0x9e,
    0x18, 0xfc, 0xcf, 0x72, 0xff, 0x3d, 0x2c, 0x56,
    0xec, 0x83, 0x5e, 0xb4, 0x0c, 0xc2, 0x48, 0xac,
    0x12, 0xd1, 0xda, 0x8b, 0xbc, 0x3e, 0x48, 0x1c,
    0xf1, 0x34, 0xda, 0x14, 0x98, 0xe2, 0xc0, 0x9f,
    0x09, 0x8b, 0x0b, 0xe6, 0xd8, 0x80, 0x73, 0xb1,
    0xa8, 0xc7, 0x1b, 0x77, 0xfd, 0x34, 0x36, 0x47,
    0xcf, 0xad, 0x42, 0x1a, 0x41, 0x11, 0xc1, 0x58,
    0xec, 0xa7, 0xc1, 0x8c, 0x67, 0xe8, 0x72, 0xd2,
    0x08, 0x1b, 0xf8, 0xb0, 0x11, 0x23, 0x1b, 0xcb,
    0x8b, 0x2c, 0xc8, 0x7e, 0x98, 0x96, 0xba, 0xf3,
    0xee, 0x86, 0x4a, 0x0f, 0xae, 0x99, 0x26, 0x79,
    0x18, 0xc4, 0xce, 0x4f, 0xba, 0x04, 0xad, 0x80,
    0x76, 0xac, 0x3d, 0x57, 0x90, 0x1f, 0x90, 0xc3,
    0x63, 0x6a, 0x1b, 0x4a, 0x8f, 0x66, 0xfc, 0x5f,
    0xf3, 0x31, 0x08, 0x41, 0xa8, 0x19, 0xb6, 0x76,
    0xa9, 0xaa, 0x57, 0x0d, 0x03, 0x0d, 0x7a, 0x9a,
    0x8b, 0x71, 0x78, 0xf6, 0x45, 0x38, 0xaf, 0xb0,
    0x7a, 0xbc, 0xe5, 0x48, 0x79, 0x6e, 0xa4, 0x7d,
    0xd7, 0xe0, 0x22, 0x09, 0x2d, 0x08, 0xf5, 0xc8,
    0x24, 0x3a, 0x57, 0x45, 0x01, 0xdc, 0x78, 0x4d,
    0x2b, 0xe9, 0xa9, 0x45, 0x18, 0x5a, 0x02, 0xca,
    0xcd, 0x37, 0x03, 0x75, 0x77, 0xd8, 0x91, 0xc3,
    0xfb, 0xfb, 0x8d, 0x9f, 0x40, 0x8d, 0xb6, 0xaa,
    0xed, 0xd7, 0x69, 0x28, 0xd4, 0x1d, 0x23, 0x58,
    0xf4, 0x94, 0xeb, 0x77, 0xdc, 0x53, 0xb6, 0x7a,
    0xfd, 0x6e, 0xc5, 0xe1, 0xa6, 0xb6, 0x2e, 0x3b,
    0x54, 0xf7, 0xbf, 0xb5, 0x5f, 0x0f, 0xf4, 0xf5,
    0xb7, 0xf5, 0xe6, 0x9c, 0xa4, 0x57, 0x15, 0xbe,
    0x42, 0x4e, 0x49, 0x0e, 0xe6, 0x9e, 0x1c, 0x57,
    0x49, 0x98, 0x9f, 0xa6, 0xc9, 0x9c, 0x20, 0xce,
    0xd5, 0x1f, 0x80, 0xed, 0xff, 0xd2, 0xfb, 0x96,
    0xc8, 0x6f, 0x5e, 0x75, 0x57, 0xe4, 0x09, 0x14,
    0x3d, 0x99, 0xa6, 0xab, 0x92, 0x70, 0xb9, 0x4f,
    0x1e, 0xa1, 0xfd, 0xce, 0xaa, 0x76, 0x67, 0x8b,
    0xce, 0x46, 0xc9, 0x48, 0x6b, 0x28, 0xeb, 0x08,
    0x55, 0x88, 0xff, 0x23, 0x5c, 0x2b, 0xd0, 0xaa,
    0x24, 0x59, 0x57, 0x73, 0xa5, 0xc5, 0x14, 0xce,
    0xa0, 0x26, 0xa5, 0x79, 0x23, 0x99, 0x29, 0x28,
```

```
    0x39, 0xcf, 0x7d, 0xa8, 0x14, 0xa9, 0x46, 0x15,
    0xb9, 0xcb, 0x80, 0x3b, 0xde, 0x76, 0x5b, 0x06,
    0x33, 0x15, 0x02, 0x8a, 0xae, 0x80, 0x49, 0x23,
    0x25, 0x2f, 0xe8, 0xf2, 0x29, 0xdc, 0x63, 0x42,
    0x7d, 0xd2, 0xa7, 0xcd, 0x60, 0x92, 0x6d, 0xe8,
    0x9b, 0x04, 0xa9, 0x53, 0x67, 0x5a, 0x89, 0x2b,
    0xc8, 0x95, 0x08, 0x1b, 0x5c, 0x82, 0xae, 0xff,
    0x39, 0xaa, 0x90, 0xa5, 0x26, 0x72, 0x95, 0xeb,
    0xd3, 0x2f, 0x42, 0x1a, 0xb0, 0xb4, 0xa4, 0xce,
    0x9e, 0xf0, 0xd1, 0x5e, 0xbb, 0x1e, 0xaf, 0x8e,
    0x0c, 0xec, 0x0a, 0xbe, 0x83, 0x82, 0x43, 0xd8,
    0x28, 0x63, 0xe2, 0xc0, 0xcb, 0x59, 0xb2, 0xd8,
    0x78, 0x3b, 0x18, 0xee, 0x09, 0x15, 0xa2, 0x35,
    0xbd, 0xab, 0x18, 0xa8, 0xd2, 0x1c, 0x57, 0x98,
    0x4f, 0xe8, 0xd7, 0xab, 0x9f, 0x81, 0xcd, 0xe6,
    0x78, 0xf8, 0x5b, 0xea, 0x0b, 0xed, 0x67, 0xc6,
    0xf1, 0x38, 0x0c, 0x1f, 0x9c, 0x19, 0x3d, 0x6c,
    0x18, 0xb1, 0x2c, 0xa9, 0x0d, 0x85, 0xcc, 0x99,
    0x37, 0xf1, 0x0a, 0xbf, 0x3b, 0x1b, 0xe0, 0x5a,
    0xc4, 0x14, 0x1b, 0xc0, 0x72, 0x5c, 0x2d, 0xc5,
    0x66, 0xde, 0x3b, 0x17, 0x3d, 0x1c, 0x5d, 0x46,
    0x1f, 0x87, 0x33, 0xe7, 0x29, 0xdf, 0x2f, 0x75,
    0x40, 0xe3, 0xf2, 0xb8, 0xd1, 0x3f, 0xfb, 0xd6,
    0xdf, 0xa3, 0x4f, 0x54, 0x72, 0xe4, 0xb6, 0xef,
    0x8b, 0xde, 0xb4, 0x5d, 0x77, 0xab, 0xb3, 0xc4,
    0x63, 0x59, 0xad, 0xe0, 0xbd, 0x95, 0x12, 0xa6,
    0xd1, 0x8e, 0x74, 0x72, 0xae, 0xd3, 0xee, 0x4b,
    0x3f, 0x67, 0xa3, 0xc0, 0xbd, 0x01, 0xd1, 0xec,
    0xfa, 0xa4, 0x70, 0x59, 0xeb, 0x28, 0x24, 0xc8,
    0xf2, 0x2e, 0xe4, 0xee, 0x5c, 0x99, 0xb7, 0xdc,
    0xef, 0xa6, 0xcd, 0xc2, 0x74, 0xf3, 0xac, 0x26,
    0xbb, 0x81, 0xd3, 0xa8, 0x32, 0xda, 0x4e, 0x92,
    0xdb, 0xf5, 0xe8, 0x3d, 0x59, 0x34, 0xd1, 0xf7,
    0xbc, 0xfd, 0xc5, 0x94, 0x71, 0x5a, 0xb7, 0xa7,
    0xb3, 0xce, 0xc9, 0x67, 0x7a, 0xdd, 0xd0, 0x0f,
    0xfb, 0xa1, 0xcf, 0xb0, 0x09, 0x31, 0x45, 0x5e,
    0x21, 0x7f, 0xab, 0x5b, 0xaf, 0x9e, 0x1e, 0xd2,
    0x69, 0x84, 0x67, 0x06, 0x61, 0x4c, 0xe6, 0x93,
    0xa1, 0x4d, 0x64, 0x06, 0xc0, 0xaf, 0x91, 0x65,
    0xf7, 0xd7, 0x19, 0x9f, 0x57, 0xe0, 0x98, 0x4d,
    0xab, 0x45, 0x71, 0x3e, 0x29, 0x7e, 0xc4, 0xc6,
    0xd8, 0x48, 0x7b, 0x64, 0x84, 0x8e, 0x22, 0x2e,
    0x24, 0x91, 0xdb, 0x90, 0x5a, 0x8a, 0x19, 0x90,
    0x10, 0xbb, 0xc7, 0x25, 0x04, 0x81, 0x04, 0x03,
    0xb5, 0x4b, 0x03, 0xe5, 0xb3, 0xa8, 0xb6, 0x46,
    0x22, 0xb2, 0xfb, 0xef, 0x28, 0xa5, 0xe1, 0x2e,
    0x68, 0xf3, 0x14, 0xb3, 0x7d, 0x9e, 0x31, 0x87,
    0xff, 0xee, 0x40, 0xa5, 0xae, 0x52, 0x6a, 0x5e,
    0xa3, 0xf7, 0xef, 0xc6, 0x2e, 0x54, 0x92, 0xba,
    0x7d, 0x8d, 0x90, 0x73, 0x7b, 0x1e, 0x7b, 0x19,
    0x6b, 0x07, 0xcb, 0xbe, 0xf5, 0x78, 0xe4, 0x9e,
    0xcd, 0x7b, 0x16, 0xab, 0x85, 0x3e, 0xd3, 0x83,
    0xa8, 0x2a, 0xdc, 0xc8, 0x07, 0x23, 0xc3, 0x75,
    0x17, 0xe0, 0xc9, 0xdf, 0xa5, 0x62, 0xe1, 0x21,
    0x54, 0x61, 0x8f, 0x04, 0xe4, 0xd1, 0xc9, 0x51,
    0xe8, 0x04, 0xc5, 0x61, 0x97, 0xe6, 0x16, 0xd6,
    0xdc, 0x3f, 0x22, 0x48, 0x60, 0x75, 0x13, 0x8d,
```

```
    0x3e, 0x1f, 0xd3, 0x50, 0xd7, 0x19, 0x03, 0xc7,
    0x68, 0xb9, 0xcb, 0x88, 0x6e, 0x6f, 0x4d, 0x02,
    0xc4, 0x3d, 0xb2, 0x91, 0x5b, 0x01, 0xe7, 0x4a,
    0x28, 0x15, 0xcb, 0xbc, 0x0e, 0xbb, 0x05, 0x28,
    0xab, 0xaf, 0x63, 0x13, 0x61, 0x8d, 0xa1, 0xb6,
    0x2f, 0x84, 0x3f, 0xa0, 0x69, 0xed, 0xf5, 0x96,
    0x7b, 0xf6, 0x4d, 0xee, 0x80, 0x73, 0xb5, 0xd1,
    0x61, 0x40, 0x6a, 0x10, 0xef, 0x6b, 0x38, 0x62,
    0xf9, 0x6a, 0xaa, 0x21, 0x12, 0x2e, 0x6e, 0x34,
    0x30, 0x5e, 0x19, 0xd3, 0x35, 0x64, 0x4c, 0x98,
    0xec, 0xe7, 0xf3, 0xc2, 0xdc, 0xcb, 0x1f, 0xdf,
    0x26, 0x2c, 0xd0, 0xbe, 0xb8, 0xd4, 0x18, 0xf3,
    0x25, 0xa0, 0x70, 0xce, 0x85, 0x6b, 0x7f, 0x47,
    0x42, 0x81, 0xb2, 0xa6, 0xed, 0x35, 0xf3, 0xa3,
    0x54, 0x40, 0xcc, 0x69, 0x64, 0x28, 0x18, 0xe2,
    0x4f, 0x4e, 0xd5, 0x04, 0xa9, 0x89, 0x8a, 0xf9,
    0xfc, 0xd1, 0xca, 0x6e, 0xee, 0x63, 0xe4, 0xfa,
    0x02, 0x04, 0x81, 0x15, 0x8e, 0x01, 0x91, 0xb7,
    0x0a, 0x8f, 0xc8, 0x32, 0x76, 0xf6, 0xfa, 0xd5,
    0x70, 0xd5, 0xcb, 0xa8, 0x9d, 0x9c, 0xd7, 0x19,
    0xb5, 0x15, 0xa0, 0x99, 0x21, 0xc0, 0xf4, 0x18,
    0xb4, 0xae, 0x22, 0x03, 0xda, 0xf4, 0xaf, 0x4f,
    0x3a, 0xef, 0x34, 0x4b, 0xf2, 0xc3, 0xc3, 0xff,
    0x04, 0xd0, 0xcc, 0xaa, 0xe3, 0x1c, 0xa4, 0xe3,
    0x6f, 0xba, 0x3d, 0xee, 0x87, 0x16, 0x35, 0xe2,
    0xaf, 0x6c, 0xcc, 0x60, 0xfd, 0xd1, 0xc2, 0x5b,
    0xf9, 0x02, 0xe8, 0x2f, 0xcd, 0x7a, 0x1f, 0x68,
    0x35, 0xd9, 0xd8, 0x3c, 0x9b, 0x99, 0xc3, 0x67,
    0xe2, 0xcb, 0xdd, 0x4b, 0x28, 0x80, 0xc2, 0xac,
    0x7d, 0xdc, 0xfb, 0x98, 0x29, 0x17, 0xf3, 0xa6,
    0x4e, 0x4e, 0xe0, 0x21, 0x1e, 0x54, 0x29, 0xc2,
    0x0a, 0xa2, 0xa4, 0x32, 0xf0, 0x91, 0x01, 0xbf,
    0xd1, 0xf1, 0x5c, 0x3f, 0xe5, 0x43, 0x06, 0x84,
    0x35, 0xe3, 0x38, 0x11, 0xcf, 0x84, 0xb2, 0x0b,
    0x89, 0x34, 0x7c, 0x2b, 0x7a, 0x05, 0x44, 0x4a,
    0xe3, 0x88, 0x0a, 0x85, 0xdb, 0xcc, 0xab, 0x25,
    0x8b, 0x4e, 0x6b, 0x7c, 0x74, 0x44, 0x07, 0xb6,
    0x52, 0x18, 0xbd, 0xc7, 0x3d, 0x6c, 0x05, 0xa7,
    0x79, 0x04, 0x49, 0x5f, 0x94, 0xd0, 0x6e, 0xbf,
    0xff, 0x11, 0x58, 0x58, 0x20, 0xf9, 0x5d, 0xcd,
    0x0d, 0x44, 0xa1, 0xe1, 0x09, 0xe6, 0x89, 0xa1,
    0xd6, 0x58, 0x9f, 0x3f, 0x3b, 0x25, 0xc5, 0x6b,
    0xf7, 0xb1, 0x92, 0x4a, 0x89, 0xe7, 0x91, 0xd9,
    0x06, 0x3c, 0x6e, 0x74, 0x18, 0x66, 0xd9, 0x6b,
    0xd9, 0xc9, 0x85, 0x7d, 0xa3, 0x81, 0x70, 0x14,
    0x90, 0x59, 0x1f, 0x98, 0xcc, 0x3b, 0x70, 0x82,
    0xd1, 0x93, 0x76, 0x87, 0xd8, 0x5a, 0xfe, 0x3a,
    0xc4, 0xd2, 0x44, 0xb0, 0xcc, 0x52, 0x91, 0xc8,
    0xfa, 0x9a, 0x94, 0x90, 0xfa, 0x8d, 0x77, 0x61,
    0x47, 0x02, 0xa6, 0xb7, 0xe4, 0x18, 0x4e, 0x0a,
    0x7b, 0x79, 0x40, 0x90, 0x32, 0x02, 0x43, 0x7e,
    0xe3, 0xdf, 0x63, 0xcb, 0xb8, 0xe7, 0x51, 0x27,
    0x5c, 0x1a, 0x7a, 0x48, 0x89, 0x1c, 0x7a, 0x54,
    0xc4, 0xc8, 0x37, 0xdb, 0xaf, 0x96, 0x4a, 0xe0,
    0x45, 0x9b, 0x4e, 0x66, 0x16, 0x6c, 0xb8, 0x8c,
    0xd1, 0x10, 0x50, 0x95, 0x25, 0x9c, 0x35, 0x35,
    0xc2, 0x1c, 0xc4, 0x63, 0x9f, 0x85, 0x57, 0x9e,
```

```
    0x37, 0x34, 0xdb, 0xfd, 0x5f, 0x7a, 0xdf, 0xc2,
    0xf9, 0x6e, 0xa9, 0x3f, 0x41, 0xa5, 0x53, 0x69,
    0x13, 0xce, 0xdc, 0x2a, 0xc4, 0x4c, 0xde, 0x43,
    0x8a, 0x6a, 0x93, 0xcb, 0xc4, 0xb8, 0x9b, 0x1d,
    0x89, 0xb3, 0x49, 0x5e, 0xb7, 0x7c, 0x51, 0xf2,
    0x8e, 0xa6, 0x5b, 0xa8, 0xe9, 0x52, 0x20, 0x49,
    0x61, 0x9e, 0xe7, 0xdf, 0x0d, 0xc0, 0xd0, 0xc8,
    0x32, 0x77, 0xb1, 0x4a, 0xe2, 0x5e, 0xe1, 0x8f,
    0xbf, 0x75, 0x77, 0xa7, 0xd1, 0x39, 0xac, 0xbb,
    0x22, 0x8b, 0x96, 0xfa, 0xc4, 0xa3, 0x92, 0x9d,
    0x5e, 0x92, 0x0d, 0xee, 0xc8, 0x5e, 0x1b, 0x6d,
    0x1e, 0x72, 0xb5, 0x49, 0x5a, 0x19, 0x70, 0x50,
    0x76, 0xc8, 0x30, 0x07, 0x0d, 0x9d, 0x76, 0x38,
    0xb1, 0x66, 0xd0, 0x76, 0x46, 0xd1, 0xf8, 0x93,
    0x34, 0xfe, 0xb2, 0x45, 0x57, 0x19, 0x58, 0x35,
    0xcd, 0xe0, 0x98, 0x4d, 0x1d, 0x75, 0x4e, 0x4e,
    0x38, 0x34, 0xa5, 0x26, 0x17, 0x93, 0xa8, 0xa3,
    0x9b, 0xa0, 0x23, 0xa4, 0xcd, 0x2f, 0x7b, 0x82,
    0x77, 0xed, 0xf6, 0x8a, 0x08, 0xbf, 0xec, 0x8d,
    0x53, 0x66, 0xd1, 0xb5, 0x3e, 0x62, 0xcd, 0x27,
    0xf8, 0x90, 0x1a, 0x1e, 0x28, 0x29, 0xf7, 0x15,
    0x4d, 0x82, 0x75, 0x71, 0xe4, 0x4a, 0xf3, 0x6d,
    0x68, 0x18, 0x17, 0x25, 0xfb, 0x98, 0x0d, 0x09,
    0xc0, 0x68, 0x5a, 0x36, 0x45, 0x9d, 0x87, 0x69,
    0x07, 0xb5, 0xf9, 0xc1, 0x8e, 0xc0, 0xa2, 0xe2,
    0xe0, 0x3c, 0x03, 0x0d, 0xc0, 0xfe, 0x6a, 0xc0,
    0x0f, 0x8d, 0x58, 0xdc, 0xcb, 0xf7, 0x30, 0x3f,
    0xf5, 0x03, 0xf2, 0x83, 0xfd, 0x61, 0x94, 0x86,
    0x7e, 0xb4, 0xae, 0x32, 0x73, 0xbc, 0xa1, 0xf0,
    0x0d, 0x05, 0x8d, 0x96, 0x7d, 0xff, 0xf8, 0xee,
    0x80, 0x28, 0x43, 0xcd, 0x16, 0xa9, 0x50, 0x10,
    0x41, 0x07, 0x8a, 0xee, 0xf7, 0xbf, 0x8f, 0x89,
    0x8c, 0xcb, 0x9b, 0x8d, 0xcd, 0xd4, 0x38, 0x56,
    0xf5, 0xec, 0xad, 0xbe, 0x77, 0x08, 0x48, 0xf6,
    0x3a, 0xda, 0x8d, 0x2f, 0xa5, 0x6a, 0xe5, 0xbf,
    0x6e, 0x92, 0xad, 0x5b, 0x30, 0xa7, 0x56, 0x6c,
    0xa9, 0xc0, 0x52, 0xf0, 0xf9, 0x6f, 0x84, 0xad,
    0x93, 0xf8, 0x1d, 0xb6, 0x3b, 0xf1, 0x61, 0xe6,
    0x0f, 0x4e, 0x52, 0x0d, 0x01, 0xf5, 0x74, 0xcc,
    0xff, 0x5c, 0xcc, 0x50, 0x83, 0x98, 0x8f, 0x8e,
    0x57, 0xa2, 0x17, 0x7f, 0xa2, 0x1b, 0x8c, 0x38,
    0x91, 0x3e, 0x9d, 0x4a, 0x76, 0xc3, 0xd0, 0xbd,
    0x11, 0x8a, 0x5f, 0xb2, 0x93, 0xad, 0xd1, 0x56,
    0x4b, 0x07, 0xee, 0xd6, 0x5d, 0x61, 0x9b, 0xde,
    0xcc, 0x59, 0x50, 0x5d, 0xa0, 0x26, 0xfa, 0x55,
    0xb7, 0x14, 0xf7, 0x4a, 0xe2, 0xa1, 0x18, 0x94,
    0x22, 0xf2, 0x5f, 0x62, 0x47, 0x0f, 0xe3, 0xeb,
    0x42, 0x54, 0x8b, 0xb1, 0xd9, 0xbe, 0xce, 0x00,
    0xc6, 0x34, 0x88, 0x5b, 0x4b, 0x09, 0x20, 0xeb,
    0xfb, 0xa8, 0x4f, 0xc2, 0x00, 0xe2, 0x8b, 0xc9,
    0x07, 0x5b, 0xa1, 0x77, 0x7e, 0x10, 0x59, 0xc9,
    0x9e, 0x9a, 0x4f, 0x5d, 0xba, 0x84, 0x5c, 0x8a,
    0xfc, 0xb6, 0x89, 0x3f, 0x09, 0x1c, 0xa9, 0x1f,
    0x22, 0xea, 0xc9, 0x31, 0x30, 0xef, 0x7b, 0xf3,
    0x60, 0xcd, 0x3d, 0xc9, 0xef, 0x4b, 0xb6, 0x19,
    0x2d, 0x41, 0xad, 0x17, 0x3c, 0x66, 0x9f, 0xdd,
    0x26, 0x48, 0x0a, 0xd1, 0x5e, 0x09, 0x22, 0x31,
```

```
    0x58, 0xe4, 0x14, 0x43, 0x21, 0x14, 0xed, 0x81,
    0x8a, 0xc8, 0xea, 0x53, 0x79, 0x01, 0xf5, 0xb0,
    0x57, 0xc7, 0xe6, 0x4b, 0xaa, 0xd1, 0x2f, 0xf5,
    0xd5, 0x38, 0x35, 0xfb, 0x8a, 0x6a, 0xf7, 0x37,
    0x65, 0x12, 0x55, 0x1b, 0xc2, 0x38, 0x97, 0x01,
    0x1f, 0x2c, 0x56, 0x55, 0x6a, 0xd0, 0xff, 0x0d,
    0x9e, 0x40, 0xcd, 0x02, 0x4a, 0x44, 0x12, 0xde,
    0xc0, 0x3d, 0xa0, 0xee, 0x66, 0x8f, 0xa6, 0xec,
    0x80, 0x12, 0x86, 0x42, 0x52, 0x4e, 0xff, 0x17,
    0x41, 0x86, 0xb4, 0x26, 0x69, 0x82, 0x00, 0x18,
    0xae, 0x3b, 0x82, 0xdc, 0x1f, 0x93, 0x8b, 0x6b,
    0x6d, 0x34, 0x5e, 0x7f, 0xb1, 0x78, 0xf1, 0x9f,
    0x99, 0x4a, 0x27, 0x68, 0xb5, 0x32, 0x57, 0x37,
    0x3d, 0xef, 0xcc, 0x1f, 0xc2, 0x4b, 0x39, 0x4c,
    0xcf, 0x16, 0x1f, 0x4e, 0x38, 0x73, 0xcd, 0xe8,
    0xae, 0xae, 0x57, 0x7f, 0x17, 0xdc, 0xc9, 0x36,
    0x61, 0x19, 0xe9, 0x44, 0x4b, 0xfe, 0x28, 0xf0,
    0xc2, 0xe5, 0x9c, 0x3c, 0xb2, 0xde, 0x57, 0xae,
    0x6a, 0x16, 0xee, 0xca, 0xcc, 0x80, 0x6a, 0xae,
    0x88, 0xd9, 0x60, 0xe4, 0xf9, 0x5b, 0x66, 0xea,
    0xbe, 0xb7, 0xb6, 0xe2, 0x60, 0x45, 0xc7, 0x42,
    0x29, 0xa0, 0x95, 0x18, 0x45, 0x2d, 0x35, 0xc1,
    0x87, 0x07, 0x2e, 0x3c, 0xee, 0x77, 0x10, 0x61,
    0x54, 0x0f, 0x5a, 0xf6, 0xf0, 0xc8, 0xce, 0x3d,
    0xee, 0xe7, 0x8e, 0x6d, 0x6c, 0x13, 0x24, 0xa1,
    0x60, 0x5a, 0x98, 0x64, 0xe6, 0x66, 0x1e, 0xd7,
    0x08, 0xac, 0x23, 0xbf, 0x58, 0xf9, 0x7b, 0x4c,
    0x4f, 0x92, 0xe1, 0x37, 0xba, 0xd9, 0xb3, 0x30,
    0x3e, 0xec, 0x68, 0xe8, 0x4e, 0x8d, 0x22, 0x76,
    0xa8, 0x96, 0x6d, 0x81, 0x12, 0x6c, 0x46, 0xdb,
    0xeb, 0xe4, 0xd5, 0xf6, 0x17, 0xf5, 0xa0, 0xb1,
    0xe3, 0x7e, 0xa6, 0xe7, 0x9b, 0xbd, 0x6b, 0xd7,
    0x3e, 0xc8, 0x7b, 0x9a, 0xde, 0x57, 0x10, 0xca,
    0x76, 0x0c, 0xf0, 0xa8, 0xcb, 0xac, 0xf2, 0x42,
    0x69, 0x80, 0x88, 0xfd, 0x72, 0xaf, 0x76, 0xc9,
    0xdd, 0x88, 0x7a, 0x9c, 0xe9, 0x39, 0x85, 0x3b,
    0x43, 0xb4, 0x53, 0xf9, 0x68, 0xcd, 0x83, 0x6d,
    0xd4, 0x27, 0x11, 0xb9, 0x66, 0x10, 0x82, 0x5b,
    0x73, 0x63, 0x36, 0x85, 0x11, 0x98, 0x74, 0xba,
    0x3d, 0x40, 0x0d, 0xef, 0x72, 0xb2, 0x34, 0x9d,
    0x46, 0x82, 0x5e, 0xc2, 0x39, 0x26, 0xbd, 0xa9,
    0xf0, 0xba, 0xe0, 0x70, 0x31, 0xc9, 0x9b, 0xf8,
    0xa4, 0x86, 0x15, 0x4f, 0x03, 0x7a, 0x87, 0xe2,
    0x5d, 0x41, 0xa9, 0x6e, 0x6c, 0x24, 0xaf, 0x94,
    0xad, 0x50, 0xf5, 0xe6, 0x08, 0x80, 0x2f, 0x04,
    0x03, 0x6e, 0x33, 0xac, 0x9d, 0xb2, 0x95, 0x58,
    0xdf, 0xf3, 0x7c, 0x38, 0x37, 0xfa, 0xe6, 0xf1,
    0x07, 0xf7, 0x4e, 0x99, 0xb2, 0x1e, 0xc5, 0xce,
    0xb2, 0x83, 0x02, 0x59, 0x18, 0xa3, 0x9d, 0x80,
    0xfd, 0x59, 0x2e, 0x74, 0x99, 0x03, 0x14, 0x70,
    0xae, 0xed, 0x74, 0x39, 0x91, 0x98, 0xfd, 0xc9,
    0x48, 0x22, 0x48, 0xba, 0x29, 0x6b, 0x1d, 0xfb,
    0xb1, 0x32, 0x09, 0x5f, 0xc5, 0x31, 0xc7, 0x2b,
    0x83, 0xba, 0xaf, 0xb0, 0x17, 0x0d, 0x4b, 0x1b,
    0x36, 0xa9, 0x4d, 0x3c, 0xb2, 0x33, 0x5d, 0x8b,
    0x95, 0xda, 0xc3, 0x63, 0x31, 0x16, 0x6d, 0xfd,
    0xf1, 0x72, 0xea, 0x9a, 0x54, 0x71, 0x4a, 0x26,
```

```
    0xe1, 0xc3, 0x58, 0x32, 0x82, 0xf0, 0xf5, 0x01,
    0xf9, 0xee, 0xac, 0x19, 0x7e, 0x2c, 0xf0, 0xb1,
    0x51, 0xee, 0x37, 0x3a, 0xdc, 0x8e, 0x80, 0xe5,
    0x54, 0xf2, 0x3d, 0x85, 0xb5, 0x29, 0x95, 0x65,
    0x7e, 0xb6, 0x19, 0xb9, 0x7b, 0xda, 0xb1, 0xad,
    0xc7, 0xeb, 0xba, 0x68, 0xf6, 0x84, 0xce, 0x56,
    0xa5, 0x76, 0xe0, 0xcd, 0x04, 0x9b, 0x44, 0x67,
    0xae, 0x52, 0x14, 0x47, 0x41, 0x13, 0xa5, 0xea,
    0xd8, 0x4b, 0xed, 0x66, 0x87, 0xb4, 0xa3, 0xe1,
    0x69, 0x3e, 0xb8, 0x96, 0x6a, 0x5f, 0x91, 0xea,
    0xd2, 0x13, 0xca, 0x31, 0x2e, 0xe6, 0xa8, 0x28,
    0xf8, 0x52, 0x84, 0x54, 0x15, 0xff, 0x5d, 0xc1,
    0xd4, 0x0a, 0x8c, 0x0f, 0x71, 0x3b, 0xe5, 0xba,
    0xaa, 0x26, 0xb9, 0x30, 0x8c, 0x3d, 0x97, 0x37,
    0x1f, 0x04, 0x75, 0x95, 0x90, 0xa5, 0x9e, 0x1f,
    0xcb, 0xd0, 0x99, 0x35, 0xdd, 0x27, 0xaf, 0x1c,
    0xbf, 0x09, 0x02, 0x8b, 0xf5, 0xbf, 0x57, 0xc0,
    0x29, 0xff, 0x54, 0xef, 0xdc, 0xe6, 0x03, 0x11,
    0x66, 0x41, 0xf8, 0x38, 0xe1, 0x2f, 0x7d, 0x9c,
    0xc2, 0xb8, 0x65, 0xc6, 0x31, 0xe1, 0xf2, 0x99,
    0xec, 0x4e, 0x1d, 0xf5, 0xa3, 0x80, 0xbc, 0xd3,
    0x10, 0x1e, 0x7d, 0xb9, 0x4e, 0x70, 0x0d, 0x3e,
    0xf8, 0x78, 0xcd, 0x78, 0x3b, 0xde, 0x1d, 0x3a,
    0xba, 0x3d, 0xc6, 0x9b, 0xca, 0xf2, 0x24, 0x1e,
    0xec, 0xf5, 0xba, 0x36, 0xf7, 0x72, 0x1d, 0x84,
    0xf0, 0x14, 0x57, 0x20, 0x55, 0xfc, 0xf6, 0xeb,
    0x9a, 0x9d, 0x53, 0x5c, 0x67, 0x0d, 0xb0, 0x00,
    0x34, 0x2d, 0xa8, 0x2e, 0x2f, 0xbc, 0x0b, 0x47,
    0xf7, 0x02, 0xf8, 0xa8, 0xa3, 0xd4, 0x9c, 0xab,
    0xc3, 0xac, 0x11, 0x9e, 0x2a, 0x4b, 0xb7, 0x96,
    0x4a, 0xfd, 0x3f, 0x27, 0x3b, 0xcc, 0xba, 0xf5,
    0xf2, 0xb5, 0xff, 0xf9, 0x3a, 0xcf, 0x16, 0x74,
    0xa8, 0x55, 0x60, 0xe6, 0xd4, 0x5a, 0x8e, 0xb7,
    0x47, 0xcd, 0x00, 0x78, 0xf0, 0x45, 0xb3, 0xaa,
    0x7f, 0x27, 0x6d, 0xc6, 0xa7, 0xca, 0xce, 0x15,
    0x98, 0x64, 0x01, 0x9d, 0x59, 0x28, 0xae, 0x17,
    0xe9, 0x7b, 0x65, 0x3f, 0x6e, 0x48, 0x2b, 0x36,
    0xeb, 0x56, 0x40, 0xe6, 0x93, 0xb7, 0xad, 0x3e,
    0x45, 0xad, 0x3d, 0x78, 0x10, 0xee, 0x93, 0xb1,
    0x04, 0xa9, 0xc6, 0x7d, 0xec, 0x88, 0xa4, 0x54,
    0x43, 0x0e, 0x55, 0x77, 0x35, 0x81, 0x69, 0x5e,
    0xb7, 0x60, 0x7a, 0x56, 0x04, 0x30, 0x8b, 0x00,
    0xf7, 0x6c, 0x86, 0x79, 0x8b, 0x42, 0x29, 0xda,
    0x1f, 0x39, 0x68, 0x84, 0x7c, 0xf0, 0xa5, 0x72,
    0xcd, 0x72, 0xaa, 0xe7, 0xa9, 0xac, 0x59, 0x7a,
    0x9b, 0x37, 0x7b, 0x46, 0x80, 0x29, 0xb2, 0x96,
    0x21, 0x32, 0xcc, 0x9a, 0xbf, 0xfa, 0xa6, 0x0f,
    0x23, 0xdf, 0xcb, 0x1e, 0x6e, 0xa8, 0xbb, 0x18,
    0x21, 0x25, 0x4b, 0xda, 0x10, 0x38, 0xf9, 0x5e,
    0x7f, 0x90, 0x75, 0x10, 0xb4, 0xa7, 0x7f, 0xcc,
    0x59, 0xbc, 0x2b, 0x63, 0x04, 0x21, 0xdb, 0x90,
    0x8e, 0x8d, 0xc5, 0xcc, 0xe6, 0xdc, 0xfd, 0xfc,
    0x1e, 0xc8, 0xec, 0x9f, 0x3f, 0x44, 0xa4, 0x6c,
    0x9c, 0x20, 0xda, 0xff, 0x34, 0xe8, 0xa9, 0xba,
    0x0d, 0x46, 0x70, 0xa8, 0x11, 0xc4, 0xb6, 0xa2,
    0x07, 0x15, 0xc6, 0x75, 0x23, 0x28, 0xd1, 0xdb,
    0x5c, 0xc6, 0x01, 0x87, 0x10, 0xb8, 0x43, 0xb3,
```

```
    0x1c, 0x4c, 0xdf, 0x1e, 0x56, 0x0b, 0x15, 0x2d,
    0x28, 0xbd, 0x2e, 0x73, 0xba, 0xae, 0x29, 0x25,
    0x0d, 0x14, 0x7e, 0x91, 0x7e, 0xba, 0x56, 0xe5,
    0xbf, 0xa1, 0x2b, 0x22, 0x5a, 0x1e, 0x49, 0xb0,
    0xcd, 0xd2, 0x62, 0xbe, 0x19, 0x2d, 0xd1, 0x76,
    0x28, 0xf0, 0x16, 0xb0, 0x11, 0x34, 0xc0, 0x15,
    0xb7, 0xfc, 0xed, 0x14, 0x78, 0xd0, 0x61, 0xa5,
    0x0c, 0x78, 0xb9, 0x72, 0x5e, 0x36, 0xa5, 0xbd,
    0xa1, 0x85, 0x01, 0x71, 0x44, 0x7d, 0xff, 0x3f,
    0x4c, 0x45, 0x53, 0x57, 0x9a, 0xa6, 0xe5, 0xe8,
    0x89, 0xa8, 0x77, 0xdd, 0xee, 0x75, 0x13, 0x92,
    0xa4, 0x88, 0x8a, 0x06, 0x4d, 0xb5, 0x13, 0xf8,
    0xee, 0x88, 0xbb, 0x54, 0xdd, 0xc0, 0x55, 0x8e,
    0x51, 0xc5, 0xfe, 0xa0, 0x0c, 0x96, 0x8f, 0x79,
    0x84, 0x52, 0x40, 0x23, 0x06, 0x03, 0x36, 0x70,
    0x13, 0xbf, 0xcd, 0xb0, 0x9e, 0x57, 0xb9, 0x5b,
    0x69, 0x8f, 0x90, 0x38, 0xab, 0xf9, 0xa1, 0x5e,
    0xcf, 0x00, 0xb9, 0x1b, 0x47, 0xf9, 0x92, 0xdb,
    0xae, 0x02, 0x26, 0x3e, 0x1f, 0x18, 0xe6, 0xe9,
    0xac, 0xc0, 0x4d, 0x2a, 0x7d, 0xa2, 0x98, 0x39,
    0x0e, 0xf9, 0xaa, 0x4d, 0xdf, 0x3f, 0xae, 0x09,
    0xdb, 0x66, 0x5a, 0xd2, 0xfd, 0x9e, 0x3c, 0xaf,
    0x0f, 0x9f, 0xe0, 0xc2, 0xf7, 0xb7, 0x30, 0x80,
    0x95, 0x0f, 0x99, 0x33, 0xdf, 0x8f, 0xa3, 0x90,
    0xf6, 0xa2, 0x4c, 0x7d, 0xb5, 0x19, 0xb4, 0x9b,
    0x3e, 0xbe, 0x20, 0x55, 0x07, 0xd6, 0x27, 0x47,
    0xd5, 0xf2, 0x77, 0xad, 0xf5, 0x3a, 0x56, 0x00,
    0x9a, 0xf9, 0xa4, 0xd2, 0xc4, 0xbb, 0xc0, 0xc4,
    0x01, 0x65, 0x7d, 0x36, 0xab, 0xf1, 0x60, 0x48,
    0x21, 0xa9, 0xe4, 0xdf, 0x1d, 0x81, 0x42, 0x04,
    0x1f, 0xf2, 0x31, 0xf2, 0xcb, 0x5b, 0xf4, 0x5a,
    0x75, 0xb3, 0x6e, 0x66, 0xc5, 0xaf, 0xfd, 0x58,
    0xa0, 0xd3, 0xec, 0xb3, 0xb0, 0xbf, 0xbe, 0x1f,
    0xb6, 0xfc, 0x5a, 0xfd, 0x96, 0x1f, 0xa4, 0x2a,
    0x02, 0x2c, 0xde, 0x1d, 0x18, 0xd6, 0x3b, 0x4d,
    0x30, 0x31, 0x41, 0x7e, 0x9b, 0xbd, 0xd2, 0xb7,
    0x70, 0x64, 0x4b, 0xd4, 0x4e, 0x05, 0xb7, 0x39,
    0xd6, 0xf3, 0xf1, 0xfa, 0xbd, 0xc4, 0xff, 0x67,
    0x68, 0x34, 0x95, 0x0d, 0xf9, 0x9c, 0x78, 0x0b,
    0xe0, 0x5e, 0x97, 0x70, 0x23, 0x0e, 0x46, 0xb7,
    0x83, 0x09, 0xf8, 0xb7, 0x6a, 0x53, 0x35, 0x18,
    0x60, 0x73, 0x5f, 0x44, 0x02, 0x6c, 0xdb, 0x41,
    0x83, 0x82, 0x2c, 0x99, 0x2d, 0xce, 0xd4, 0xb3,
    0xd2, 0x3e, 0x26, 0xd7, 0x6d, 0xbe, 0xba, 0x21,
    0x9d, 0x34, 0xc0, 0x2c, 0x66, 0x52, 0x42, 0x5e,
    0x05, 0x8e, 0xe3, 0x22, 0xd2, 0xe2, 0xf4, 0x3e,
    0x62, 0x76, 0x66, 0xfd, 0x9f, 0xb3, 0xd7, 0x0f,
    0x40, 0xe2, 0x1b, 0xeb, 0x0c, 0x6f, 0x3c, 0x27,
    0xaa, 0xc5, 0xa0, 0x5d, 0xe6, 0x2a, 0xf0, 0x54,
    0xb9, 0x82, 0x2a, 0xe9, 0xc9, 0x17, 0xea, 0xb6,
    0x6a, 0x95, 0x21, 0x45, 0x9a, 0x91, 0x94, 0xd1,
    0x74, 0x4c, 0x8b, 0xc1, 0xf2, 0xa2, 0x6c, 0x42,
    0x37, 0xa2, 0x85, 0x8e, 0x6a, 0xa9, 0xe0, 0x67,
    0x80, 0x76, 0x0c, 0x16, 0x7d, 0x35, 0x55, 0x2b,
    0x98, 0xdf, 0xe3, 0x4a, 0xef, 0x00, 0x5d, 0x54,
    0x25, 0x6e, 0x8d, 0xa6, 0x73, 0xee, 0x21, 0x43,
    0xd7, 0x58, 0x48, 0x06, 0xee, 0x52, 0x9d, 0x5b,
```

```
    0x35, 0x10, 0x78, 0x0e, 0x20, 0xef, 0xb9, 0xc2,
    0x83, 0x5e, 0xd8, 0xcd, 0x65, 0xd6, 0x22, 0xc4,
    0x61, 0xf8, 0x90, 0xd1, 0x28, 0xb6, 0xea, 0x46,
    0x5a, 0xa3, 0x20, 0x42, 0x73, 0x82, 0x71, 0xef,
    0xc1, 0xc2, 0x44, 0x90, 0x3b, 0xa5, 0x01, 0x27,
    0x02, 0x6d, 0xa1, 0x8c, 0x72, 0xba, 0xe1, 0xdf,
    0x01, 0x01, 0xd7, 0xc6, 0x1e, 0xe6, 0xc1, 0xe7,
    0x69, 0x80, 0x96, 0xf5, 0xbe, 0x55, 0x5e, 0x10,
    0x3e, 0x20, 0x2b, 0x32, 0xe4, 0x04, 0x93, 0x86,
    0xaf, 0xa3, 0x50, 0xf8, 0x0b, 0xec, 0x48, 0x4c,
    0xbf, 0x94, 0x4a, 0x4c, 0x7e, 0xb9, 0xc5, 0x21,
    0x2a, 0x28, 0xfe, 0x95, 0x01, 0x9a, 0x51, 0x11,
    0xce, 0x7a, 0x48, 0x5a, 0x05, 0x6c, 0x68, 0xbb,
    0x07, 0x64, 0x8b, 0x95, 0xc7, 0x70, 0x51, 0xa1,
    0x66, 0x29, 0xc7, 0x3d, 0xf7, 0xf5, 0xb7, 0x69,
    0x15, 0xb1, 0xf9, 0x28, 0x18, 0xd7, 0x42, 0xa1,
    0x64, 0xe1, 0x58, 0x18, 0x3d, 0x21, 0x3c, 0xb3,
    0x8b, 0x3c, 0x7e, 0xa4, 0x1f, 0xa9, 0x5b, 0x56,
    0xa7, 0xd8, 0x3d, 0x3e, 0x15, 0xde, 0x5e, 0xe8,
    0x59, 0x22, 0x21, 0xc7, 0x88, 0x13, 0x06, 0xbe,
    0xaf, 0x37, 0x0c, 0x3b, 0x58, 0xd5, 0x76, 0x1e,
    0xc8, 0x94, 0xb0, 0x8e, 0x08, 0xea, 0x4d, 0x57,
    0x53, 0x58, 0x1c, 0xd8, 0xc6, 0x33, 0xff, 0xe0,
    0x40, 0x21, 0xb4, 0xe9, 0x40, 0xd6, 0x8f, 0x98,
    0xd7, 0x9e, 0xc1, 0x06, 0x16, 0xfa, 0x52, 0x49,
    0xdc, 0x97, 0x57, 0x5b, 0x16, 0x9a, 0x2f, 0xd1,
    0xc2, 0x1f, 0xc3, 0x96, 0x20, 0x25, 0x7c, 0xee,
    0x27, 0xf3, 0xe1, 0x87, 0x59, 0x5d, 0x0d, 0xa2,
    0x3a, 0x16, 0xe9, 0x4f, 0x39, 0xc4, 0x68, 0x09,
    0xe3, 0x7d, 0x0f, 0xb2, 0x15, 0x90, 0xc5, 0x9f,
    0xae, 0x29, 0xbb, 0x36, 0x14, 0x91, 0x79, 0x03,
    0x00, 0xf3, 0x2f, 0x66, 0xb4, 0x07, 0x01, 0x26,
    0xde, 0x85, 0xfe, 0x7d, 0xbf, 0x42, 0x34, 0xde,
    0x6f, 0xc7, 0x2e, 0xf6, 0x34, 0x83, 0xa7, 0x40,
    0x5a, 0xa4, 0x35, 0x60, 0x21, 0x17, 0x9b, 0xf7,
    0xcc, 0x4f, 0x6a, 0x9c, 0xf9, 0x98, 0xbe, 0xde,
    0x92, 0xf9, 0x40, 0xc9, 0x32, 0x6c, 0xa9, 0xc5,
    0xfe, 0x79, 0x49, 0x4e, 0x12, 0x48, 0xdc, 0x6a,
    0xa3, 0xa7, 0x11, 0x4a, 0xe4, 0x61, 0x1e, 0x5a,
    0x71, 0x70, 0x3e, 0x2a, 0x1a, 0x80, 0x1f, 0xfb,
    0xdc, 0xa0, 0x54, 0xda, 0xf8, 0xc5, 0xbc, 0xa2,
    0x21, 0x06, 0x19, 0xfe, 0xa4, 0x32, 0x54, 0xc6,
    0x70, 0x39, 0x2b, 0x83, 0x7f, 0x0b, 0x23, 0xbc,
    0x76, 0x1b, 0x02, 0x1f, 0x1c, 0xf3, 0xee, 0x79,
    0x17, 0x55, 0xb9, 0xc0, 0x14, 0x84, 0x5c, 0x4f,
    0x40, 0xea, 0x1f, 0xc2, 0x74, 0xf0, 0xbc, 0xa9,
    0xc5, 0x5b, 0xdb, 0x6c, 0x2b, 0xfa, 0xb8, 0xe9,
    0xf3, 0x9e, 0x67, 0x18, 0x2f, 0xf7, 0x44, 0x41,
    0x06, 0xfb, 0xb9, 0x91, 0x4b, 0x74, 0xd7, 0x17,
    0x81, 0xdf, 0x0c, 0x0f, 0x1c, 0x04, 0xc1, 0x45,
    0x1d, 0xa4, 0xa1, 0x3e, 0x18, 0xf9, 0xc7, 0x91,
    0xfc, 0x3f, 0x29, 0x0a, 0xcf, 0x44, 0x11, 0xde,
    0xa5, 0x31, 0x7b, 0xea, 0x72, 0x1e, 0x69, 0xb9,
    0x2b, 0xdd, 0xc0, 0x3c, 0x94, 0xb8, 0x93, 0x5c,
    0x9e, 0xa0, 0xd7, 0xa5, 0x1c, 0x5b, 0x71, 0xa7,
    0x3c, 0xbc, 0x33, 0x6d, 0x6e, 0x7d, 0xb7, 0x32,
    0x54, 0x17, 0x1c, 0xd2, 0x6b, 0x1a, 0x03, 0x3b,
```

```
    0xfa, 0xf3, 0x43, 0x9e, 0x75, 0x50, 0xba, 0x77,
    0xc1, 0x63, 0x8d, 0x84, 0x33, 0xea, 0xd2, 0xcd,
    0xe7, 0xa2, 0xf4, 0xe7, 0xc1, 0x1c, 0xcc, 0xb7,
    0xea, 0x38, 0x64, 0xf9, 0x0d, 0x33, 0xa0, 0x73,
    0x35, 0x78, 0x2e, 0xef, 0x42, 0xdd, 0x13, 0x8c,
    0x4d, 0x98, 0x95, 0x3d, 0xb6, 0x0d, 0xdd, 0x26,
    0x95, 0x33, 0xcc, 0xc7, 0x65, 0x08, 0xfd, 0x53,
    0x8c, 0xa5, 0xe5, 0x66, 0xb6, 0x2c, 0x86, 0xd6,
    0x07, 0xf5, 0xee, 0x14, 0xd9, 0x02, 0x86, 0xeb,
    0x92, 0xf2, 0x1e, 0xec, 0x90, 0xf3, 0x23, 0xd1,
    0x9d, 0x13, 0x45, 0xd2, 0x4b, 0x91, 0x2a, 0x8e,
    0x39, 0x9f, 0x96, 0xf4, 0x20, 0xb6, 0x81, 0x38,
    0x79, 0x35, 0x98, 0x4a, 0x64, 0x85, 0xaf, 0xff,
    0x04, 0x4e, 0x28, 0x10, 0x46, 0x80, 0x76, 0x22,
    0x66, 0x26, 0x02, 0x46, 0x45, 0x15, 0xf8, 0x92,
    0x23, 0xdc, 0x06, 0x41, 0x0a, 0x85, 0xed, 0x1f,
    0xd3, 0x51, 0x5c, 0xa4, 0xad, 0xa6, 0x21, 0xa9,
    0x56, 0x0c, 0x4b, 0xbe, 0xc8, 0xe1, 0xc1, 0xcc,
    0xd6, 0x72, 0x6f, 0xc7, 0x97, 0x50, 0x87, 0x9d,
    0x67, 0x24, 0x75, 0x00, 0x9c, 0x62, 0x2c, 0xad,
    0x7d, 0x2d, 0x12, 0x47, 0x71, 0x62, 0xc2, 0xb0,
    0x63, 0xdc, 0x9a, 0x6b, 0xea, 0x80, 0xc6, 0x9a,
    0x47, 0x6b, 0xa4, 0x45, 0x1c, 0xb4, 0xbf, 0x82,
    0xd9, 0x26, 0x08, 0x03, 0x46, 0x25, 0x65, 0xd2,
    0x70, 0xd6, 0x07, 0xd0, 0x3e, 0x77, 0xc0, 0x0c,
    0x4a, 0x9f, 0x34, 0xfd, 0xcc, 0x07, 0x7b, 0xec,
    0xcf, 0x01, 0xa8, 0xc4, 0x2e, 0x79, 0x43, 0x76,
    0xaa, 0x0e, 0xc2, 0xe4, 0xa3, 0x2d, 0xb3, 0x7e,
    0x29, 0x5d, 0xf9, 0x5e, 0x10, 0xb1, 0xce, 0xbd,
    0x21, 0x1a, 0x0d, 0x15, 0x80, 0x74, 0xb2, 0x5e,
    0x0a, 0x15, 0x49, 0x9b, 0x51, 0xbc, 0xc5, 0x75,
    0x84, 0x80, 0xbf, 0xdb, 0x0a, 0x45, 0xea, 0xda,
    0x86, 0x27, 0x93, 0x90, 0x2b, 0x3f, 0xba, 0xa1,
    0xfb, 0xe1, 0x16, 0xf8, 0x9e, 0x5b, 0xf4, 0x8b,
    0x8a, 0x8a, 0xdf, 0x3f, 0xda, 0xe3, 0x0e, 0xc4,
    0x8c, 0x49, 0x4b, 0x53, 0xc8, 0x7c, 0x49, 0xa0,
    0xa0, 0xa3, 0x2b, 0x06, 0x03, 0xed, 0x20, 0x1b,
    0xda, 0x4c, 0xf7, 0x61, 0xc9, 0x98, 0x3c, 0xd8,
    0xc7, 0x94, 0x6d, 0xf2, 0x98, 0xf8, 0x0f, 0xdc,
    0x26, 0x60, 0x0e, 0x48, 0xea, 0x73, 0x82, 0xe4,
    0xd3, 0x07, 0x21, 0x02, 0x42, 0x9e, 0x98, 0x0b,
    0xa9, 0xa3, 0x05, 0x19, 0x28, 0x7e, 0x7c, 0x7d,
    0x47,
  },
  &case10
};
```

C.3 XCB-AES test vectors

The security analysis of XCB that was published in Selected Areas in Cryptography 2007 [B3] makes use of a fact that relates the internal variables *F* and *C* (as defined in 3.1, Theorem 3, see [B3]). This relation is used as a consistency check on an implementation, since the relation holds for each invocation of the XCB algorithm. Because the proof of security makes use of the fact that the relation holds, such a check connects the security analysis to the validation of the implementation, and thus provides added confidence of correctness. The test cases in this document have been verified against this consistency check.

Table C.3 shows 11 test vectors referred to as “case 10” through “case 0.” Case 1 tests XCB-AES-256, while the others test XCB-AES-128 with different combinations of plaintext and associated data lengths.

Table C.3—XCB-AES test cases

```
test_case_t case10 = {
  /* key */
  {
    0xa9, 0x55, 0xec, 0x89, 0xee, 0x6e, 0x0f, 0xf5,
    0xe5, 0x30, 0x34, 0xc5, 0x89, 0x1c, 0x4e, 0x97,
    0x68, 0x30, 0x31, 0x2a, 0x3a, 0x94, 0xb1, 0xe8,
    0x5e, 0x30, 0xeb, 0xc6, 0x34, 0x95, 0x97, 0xea,
  },
  /* bytes in key */
  32,
  /* plaintext */
  {
    0x6e, 0xc7, 0xe1, 0x66, 0x5f, 0x80, 0x6d, 0xf4,
    0xbc, 0xbd, 0x4a, 0x4c, 0x10, 0xa0, 0x6b, 0xd8,
    0x7b, 0xfb, 0x06, 0xf4, 0x17, 0x8a, 0xe5, 0x18,
    0x70, 0x6a, 0x1d, 0x71, 0x5f, 0x44, 0x8b
  },
  /* bytes in plaintext */
  31,
  /* associated data */
  { },
  /* bytes in associated data */
  0,
  /* ciphertext */
  {
    0xfb, 0x9c, 0x5b, 0xfb, 0x11, 0xc5, 0x75, 0x28,
    0x47, 0x64, 0xaa, 0x81, 0xba, 0x18, 0x90, 0x6f,
    0x2d, 0x66, 0xf5, 0x3a, 0x52, 0x3a, 0xd4, 0xfc,
    0x2f, 0x23, 0x53, 0xa4, 0x8f, 0x0b, 0x6f
  },
  NULL
};


test_case_t case9 = {
  /* key */
  {
    0xc9, 0x9a, 0xb8, 0x97, 0xad, 0xdd, 0xcc, 0xca,
    0xb8, 0x4e, 0x0d, 0xf4, 0xea, 0xfc, 0x93, 0xa4,
    0xf7, 0x60, 0xf3, 0x92, 0x69, 0x64, 0x1c, 0x19,
    0xe5, 0x92, 0x9b, 0x71, 0x2d, 0xd3, 0xd0, 0x79,
  },
  /* bytes in key */
  32,
  /* plaintext */
  {
    0xb1, 0x13, 0x50, 0x83, 0x81, 0x22, 0x96, 0x8d,
    0xbb, 0xf2, 0xaa, 0xe6, 0x9b, 0xfd, 0xf5, 0xdb,
    0x5b, 0xff, 0x16, 0x6f, 0xe7, 0x14, 0x03, 0x9b,
    0xd3
```

```
  },
  /* bytes in plaintext */
  25,
  /* associated data */
  {
    0xb7, 0xd6, 0xbb, 0x6d, 0x90, 0x35, 0x22, 0x08,
    0x02, 0x69, 0xb3, 0xa5, 0x75, 0x61, 0x5a, 0xf8,
    0xc7, 0x5a, 0x52, 0xa4, 0x82, 0x86, 0xe3, 0x46,
    0x15, 0xfc, 0x6c, 0x28, 0x9b, 0x09, 0x57,
  },
  /* bytes in associated data */
  31,
  /* ciphertext */
  {
    0x02, 0x15, 0x47, 0x5a, 0xec, 0xfc, 0xe0, 0x55,
    0x00, 0xc4, 0xcd, 0xc9, 0x06, 0x8c, 0xbb, 0x65,
    0xb6, 0x6b, 0x27, 0x0e, 0xff, 0xa2, 0xe3, 0x08,
    0x9d
  },
  &case10
};

test_case_t case8 = {
  /* key */
  {
    0x3b, 0xb9, 0x6b, 0xd5, 0x0b, 0x91, 0xa7, 0xd8,
    0x37, 0x84, 0x45, 0x24, 0x26, 0x2f, 0xef, 0x97,
    0xe0, 0x41, 0x2c, 0xbd, 0x64, 0xa3, 0x91, 0xc1,
    0xd3, 0x93, 0xc1, 0x33, 0x11, 0xf1, 0x9f, 0x86,
  },
  /* bytes in key */
  32,
  /* plaintext */
  {
    0xf0, 0xae, 0x13, 0x92, 0x99, 0xc1, 0xaf, 0x3d,
    0xd0, 0xe5, 0xa0, 0x4b, 0xe3, 0x2c, 0xd3, 0xe3,
    0x93
  },
  /* bytes in plaintext */
  17,
  /* associated data */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x58, 0xe3, 0x6c, 0xeb, 0xc7, 0x41, 0x17, 0x28,
    0xc1, 0x5b, 0xe4, 0xaf, 0xad, 0x3d, 0xfd, 0x0f,
    0x18
  },
  &case9
};

test_case_t case7 = {
```

```
  /* key */
  {
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f,
  },
  /* bytes in key */
  16,
  /* plaintext */
  {
    0x08, 0x47, 0x1e, 0x46, 0x29, 0x45, 0xa7, 0x41,
    0x54, 0x0f, 0xaa, 0x16, 0xf0, 0x1e, 0x42, 0x1b,
    0x7f, 0xa4, 0x3e, 0x0d, 0x1f, 0x99, 0xf6, 0xa0,
    0x1f, 0x71, 0x26, 0xf9, 0x8a, 0x3f, 0xc9, 0x6a,
    0xd6, 0x8b, 0xf8, 0x6e, 0xa8, 0xd7, 0x2a, 0xab,
    0x5d, 0x98, 0x7d, 0x08, 0x54, 0xea, 0x72, 0xfe,
    0xa7, 0x64, 0x3c, 0x65, 0x84, 0x33, 0xdd, 0x5e,
    0x31, 0xb4, 0x06, 0x70, 0xc6, 0xd6, 0x9d, 0x1b,
    0x4c, 0xe3, 0xac, 0x9d, 0x9f, 0x5f, 0x73, 0xc6,
    0x91, 0x8a, 0xeb, 0x8d, 0x4c, 0x2d, 0xad, 0xbe,
    0x12, 0xe6, 0xd0, 0xc7, 0x2f, 0x4c, 0xa9, 0x1e,
    0x66, 0xc6, 0xbe, 0xbd, 0x32, 0xf0, 0x09, 0x48,
    0x65, 0x81, 0xda, 0x90, 0x18, 0xa7, 0x4b, 0x9c,
    0x7e, 0x28, 0x8f, 0xb1, 0x8f, 0xd6, 0x09, 0x00,
    0xa4, 0x44, 0x8f, 0xab, 0xea, 0xd7, 0x3d, 0x13,
    0xcb, 0x24, 0x83, 0xfb, 0xc8, 0xfb, 0xdf, 0xe9,
    0x30, 0xa1, 0x38, 0x90, 0x55, 0x5c, 0xaa, 0x88,
    0xf4, 0xac, 0xdd, 0x5a, 0x3e, 0x51, 0x59, 0xe5,
    0xa6, 0x46, 0x7e, 0xc7, 0xef, 0x05, 0x23, 0x95,
    0x30, 0x14, 0xe6, 0xde, 0x79, 0x6c, 0xce, 0x7d,
    0x4f, 0xcd, 0x14, 0xb0, 0x67, 0x7a, 0x2d, 0x8e,
    0x50, 0x9f, 0x55, 0xc8, 0x14, 0xed, 0x12, 0xcd,
    0x75, 0x5c, 0xd8, 0xac, 0xb7, 0xbb, 0x12, 0x66,
    0xb4, 0xd7, 0x25, 0xe2, 0x50, 0x55, 0xe4, 0xd3,
    0x60, 0xb7, 0xcd, 0x31, 0xab, 0xdd, 0x5f, 0x42,
    0x92, 0x7a, 0x4c, 0x11, 0x16, 0x30, 0x5f, 0xea,
    0x7e, 0xcb, 0xac, 0x5d, 0xc4, 0x7f, 0xf2, 0xf3,
    0x30, 0xef, 0x10, 0x8d, 0xc8, 0x93, 0xf7, 0xbe,
    0xcd, 0x6e, 0xea, 0xa3, 0x95, 0x74, 0xdb, 0x1e,
    0xe8, 0x42, 0xea, 0xab, 0x10, 0xf1, 0x7c, 0x29,
    0x93, 0x1f, 0x92, 0x52, 0xc1, 0x0c, 0x40, 0x2c,
    0xaa, 0x00, 0xe8, 0x77, 0x2d, 0x54, 0x11, 0x1a,
    0xba, 0x50, 0x6e, 0x4f, 0xef, 0x24, 0x7b, 0x58,
    0xcb, 0x6a, 0xa2, 0xfc, 0xbb, 0xc4, 0xef, 0x91,
    0xc4, 0x04, 0x5d, 0xde, 0x51, 0x32, 0xda, 0x81,
    0x12, 0x12, 0x7c, 0xa4, 0xb0, 0x0b, 0x9c, 0xa9,
    0xa4, 0x28, 0x29, 0xa4, 0xd3, 0x9a, 0xaf, 0x2b,
    0xc1, 0x27, 0xd9, 0xe6, 0x9e, 0x92, 0x4f, 0x01,
    0x69, 0x29, 0xf9, 0x5f, 0x54, 0x68, 0xbe, 0x6f,
    0xc7, 0x41, 0x58, 0xe7, 0x0d, 0xa7, 0x9c, 0x74,
    0x83, 0x54, 0xab, 0x11, 0x81, 0xee, 0xbd, 0x77,
    0x47, 0xf8, 0xfb, 0x44, 0x08, 0x72, 0xd4, 0xb4,
    0xfb, 0xa2, 0x11, 0xfb, 0x4c, 0x00, 0x9a, 0xf0,
    0xd4, 0x1a, 0xc8, 0x13, 0x44, 0x11, 0x20, 0xb9,
    0x62, 0xde, 0x53, 0x01, 0xdd, 0x54, 0x4e, 0x0c,
    0x0b, 0x1a, 0xd4, 0x3f, 0x82, 0x9f, 0x76, 0xa5,
    0x1b, 0x33, 0x1c, 0xd4, 0x26, 0x51, 0xb6, 0xa2,
    0x26, 0x28, 0x42, 0xb9, 0x0c, 0xd2, 0x93, 0x24,
```

```
    0x18, 0xd8, 0xb6, 0x70, 0x75, 0x2a, 0x99, 0x25,
    0xd2, 0xfb, 0x80, 0xfa, 0x25, 0x23, 0xb4, 0x22,
    0x21, 0x21, 0xd0, 0x09, 0x99, 0x7e, 0xf2, 0x22,
    0x3a, 0xca, 0x4b, 0x12, 0xe6, 0x28, 0x05, 0x0d,
    0xce, 0x8d, 0x0a, 0x6b, 0xdc, 0xd5, 0x47, 0x49,
    0xe0, 0xda, 0x58, 0xf3, 0xfc, 0xa5, 0x63, 0x91,
    0xb5, 0x60, 0x2b, 0x5b, 0xbb, 0x13, 0xd0, 0xf1,
    0x2b, 0x1c, 0xd3, 0x0b, 0x45, 0xb6, 0xa7, 0x62,
    0x32, 0xdc, 0x27, 0xab, 0x81, 0x97, 0x1f, 0xab,
    0xdc, 0xc7, 0x5a, 0xee, 0x7b, 0xb6, 0x8b, 0xf9,
    0x35, 0x95, 0x55, 0xe2, 0x04, 0x8c, 0xd4, 0x4b,
    0x8e, 0x7a, 0xdb, 0x89, 0x52, 0xe2, 0xf0, 0xfa,
    0x3b, 0xda, 0x38, 0xbc, 0xa6, 0x49, 0x72, 0x4a,
    0x5f, 0x1d, 0x0a, 0xac, 0x41, 0x31, 0x0d, 0x75,
    0x78, 0xa6, 0x17, 0x48, 0x88, 0x82, 0xab, 0x66,
    0x3f, 0x46, 0x26, 0x19, 0x11, 0xe4, 0xb8, 0x41,
    0x27, 0xf3, 0x70, 0x62, 0x3b, 0x9f, 0xf6, 0x2e,
  },
  /* bytes in plaintext */
  520,
  /* associated data */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x28, 0xb0, 0xec, 0x43, 0x2f, 0x39, 0x7f, 0x1b,
    0x1a, 0xe9, 0x8e, 0x45, 0x86, 0xd2, 0x92, 0x66,
    0xae, 0x7e, 0x59, 0x78, 0x7c, 0x2d, 0x8e, 0x8b,
    0x3f, 0x3f, 0x1c, 0x10, 0xda, 0xfc, 0x7e, 0x63,
    0x13, 0x21, 0xec, 0x09, 0xe7, 0xa4, 0x7a, 0x04,
    0x92, 0xf1, 0xfb, 0x52, 0xff, 0x11, 0x23, 0xd4,
    0x96, 0xaf, 0xf0, 0xad, 0xbc, 0xb9, 0x32, 0x1c,
    0x9b, 0xd2, 0x91, 0x74, 0xc4, 0x78, 0x2b, 0x28,
    0xb1, 0x18, 0x92, 0x77, 0x72, 0x96, 0xd3, 0x0c,
    0xbc, 0xf0, 0x4f, 0x6e, 0x4f, 0x7a, 0xe6, 0x1a,
    0xc0, 0xa8, 0x6a, 0x06, 0x4c, 0xe9, 0xec, 0xe8,
    0x8b, 0x3a, 0x6d, 0x32, 0xd1, 0x79, 0xba, 0xca,
    0x91, 0x66, 0xcd, 0x15, 0xc5, 0xf1, 0x68, 0x7e,
    0x88, 0x9a, 0x1e, 0xe4, 0x0b, 0x32, 0x78, 0x3b,
    0x02, 0xdd, 0xfd, 0x50, 0x0b, 0x6c, 0xd4, 0x96,
    0xba, 0x1f, 0x5d, 0x7b, 0x6e, 0xd6, 0xfd, 0xee,
    0xfd, 0xc8, 0xc3, 0x6c, 0xa3, 0x81, 0x8b, 0x51,
    0x60, 0xb5, 0x58, 0x82, 0xc6, 0x16, 0x58, 0x03,
    0xdb, 0xbe, 0xe9, 0x5e, 0x12, 0xb5, 0xe2, 0xfd,
    0x4a, 0x0a, 0xfd, 0x5d, 0x84, 0x50, 0xd0, 0x98,
    0x3e, 0x30, 0xdb, 0x63, 0x18, 0x1f, 0x9a, 0x2a,
    0x3c, 0xc5, 0x16, 0xf2, 0x07, 0x59, 0x6e, 0xf5,
    0xee, 0x92, 0x7a, 0xfb, 0xf1, 0x41, 0xf0, 0xc5,
    0x5b, 0x0b, 0x08, 0x13, 0xe2, 0x99, 0x5b, 0x7c,
    0x4c, 0x13, 0xc0, 0x22, 0xe0, 0xba, 0x00, 0x42,
    0x27, 0x8b, 0x13, 0x32, 0x39, 0x1d, 0xb8, 0x9c,
    0x5d, 0xec, 0x68, 0x2f, 0xcd, 0xba, 0xdf, 0xba,
    0x6c, 0x01, 0x83, 0x25, 0x48, 0x47, 0x8f, 0x60,
```

```
    0x06, 0x21, 0x98, 0xa9, 0x5c, 0x85, 0xa3, 0xc8,
    0xf6, 0x33, 0x75, 0x3d, 0xc1, 0xe2, 0x9a, 0xc5,
    0x60, 0xf5, 0xf5, 0xf8, 0x1d, 0x9e, 0xaa, 0x24,
    0x00, 0x76, 0x65, 0x6b, 0x84, 0xe1, 0xd9, 0x20,
    0xb9, 0xd9, 0x68, 0xee, 0xb8, 0x4c, 0x74, 0x1a,
    0x22, 0x54, 0xe5, 0x11, 0x2c, 0x33, 0x92, 0xfb,
    0xd4, 0xf9, 0xb2, 0xdd, 0x30, 0x75, 0x2b, 0xf2,
    0x69, 0xef, 0x30, 0xa3, 0xca, 0x5c, 0x67, 0x35,
    0x6e, 0x4e, 0x53, 0xd9, 0xda, 0x6a, 0x1b, 0x99,
    0x55, 0x38, 0x1f, 0x85, 0x49, 0x1e, 0x52, 0xaa,
    0xdc, 0x38, 0xd8, 0x69, 0x61, 0xec, 0x53, 0x47,
    0xa7, 0x24, 0x04, 0xfc, 0x50, 0xd7, 0x33, 0x11,
    0xd8, 0x20, 0x00, 0x86, 0x98, 0x3e, 0x50, 0x35,
    0xff, 0x02, 0xb1, 0xf8, 0xf1, 0x44, 0xea, 0xef,
    0x31, 0x75, 0x12, 0x3a, 0xf4, 0x97, 0x0f, 0xc7,
    0x7e, 0x76, 0x91, 0xce, 0xe4, 0x50, 0x1d, 0x94,
    0x90, 0x69, 0xd6, 0x11, 0x6b, 0xf1, 0xb3, 0x01,
    0x2e, 0xac, 0x51, 0x07, 0x36, 0xc0, 0x9c, 0xfc,
    0x63, 0x6d, 0x01, 0x64, 0xf6, 0x9f, 0x52, 0x53,
    0xf4, 0xb4, 0x16, 0x2c, 0x5e, 0x55, 0x98, 0xcb,
    0x7b, 0x0f, 0x95, 0xff, 0xe4, 0xc0, 0x78, 0x97,
    0x1b, 0xe5, 0x49, 0x52, 0x0d, 0xec, 0x65, 0x5d,
    0xd6, 0x1d, 0x36, 0xcc, 0xa9, 0xd2, 0x6b, 0xaa,
    0x02, 0xb1, 0x8c, 0xed, 0x48, 0xfb, 0xee, 0xb4,
    0xb8, 0x42, 0xc0, 0x45, 0xc3, 0xc1, 0x18, 0x81,
    0xdc, 0x83, 0x76, 0xc5, 0xda, 0xfc, 0x82, 0xac,
    0xc6, 0xda, 0x45, 0x3a, 0xd3, 0xa1, 0x21, 0x39,
    0xab, 0x0f, 0x0f, 0x6d, 0xd7, 0xdf, 0x3b, 0x1e,
    0xe4, 0xaa, 0x71, 0x42, 0x8a, 0x19, 0xff, 0x97,
    0x31, 0x92, 0xeb, 0xd6, 0x0d, 0x6d, 0xe6, 0x98,
    0x84, 0xff, 0x99, 0xe9, 0x0d, 0xea, 0x4e, 0x5f,
    0xc0, 0xab, 0x0a, 0xa6, 0x0d, 0x96, 0x7d, 0x60,
    0x0b, 0xdd, 0x25, 0x9d, 0x5d, 0x63, 0xb3, 0xb9,
    0xd4, 0x85, 0x9e, 0xf7, 0x5d, 0x3d, 0xbd, 0xe2,
    0xd1, 0x4f, 0x17, 0x66, 0x07, 0xff, 0x3c, 0x1d,
    0xe5, 0xf6, 0x28, 0xc2, 0xfc, 0x65, 0x5f, 0x33,
    0x32, 0x29, 0xf7, 0x48, 0x12, 0x27, 0x98, 0xe3
  },
  &case8
};


test_case_t case6 = {
  /* key */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x16, 0x16, 0xdd, 0xa6
  },
  /* bytes in key */
  16,
  /* plaintext */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
  },
  /* bytes in plaintext */
```

```
  24,
  /* associated data */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x70, 0x13, 0xfd, 0xe3, 0xc3, 0x9f, 0xa1, 0xa4,
    0x3f, 0x5a, 0xb4, 0x34, 0x5a, 0xbf, 0xe5, 0xd9,
    0xcf, 0x80, 0x85, 0xf8, 0x7e, 0xb3, 0x11, 0x89,
  },
  &case7
};

test_case_t case5 = {
  /* key */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x16, 0x16, 0xdd, 0xa6
  },
  /* bytes in key */
  16,
  /* plaintext */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00
  },
  20,
  /* associated data */
  { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x70, 0x13, 0xfd, 0xe3, 0xdb, 0x56, 0x19, 0xbf,
    0xa4, 0xed, 0x25, 0x6d, 0xb4, 0x44, 0x15, 0x68,
    0x7a, 0xa4, 0x50, 0x3f
  },
  &case6
};

test_case_t case4 = {
  /* key */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0xf3, 0x24, 0x6b, 0x19,
  },
  /* bytes in key */
  16,
  /* plaintext */
  {
```

```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in plaintext */
  16,
  /* associated data */
  {
    0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x28, 0x2a, 0x71, 0x43, 0x39, 0xae, 0x66, 0x8c,
    0x3c, 0x20, 0x2a, 0xca, 0x9c, 0x71, 0xe0, 0x0b,
  },
  &case5
};

test_case_t case3 = {
  /* key */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x16, 0x16, 0xdd, 0xa6,
  },
  /* bytes in key */
  16,
  /* plaintext */
  {
    0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /*  32 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /*  64 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /*  96 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 128 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 160 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 192 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 224 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 256 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 288 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 320 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 352 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 384 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 416 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 448 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 480 */
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00  /* 512 */
  },
  /* bytes in plaintext */
  512,
  /* associated data */
  {
    0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0xbf, 0x2c, 0x04, 0x93, 0xbb, 0xb4, 0xbd, 0x55,
    0xcc, 0x11, 0xc0, 0x3d, 0xd9, 0x25, 0x1b, 0xe5,
    0x83, 0x79, 0x9f, 0x9d, 0xba, 0xcf, 0x23, 0x16,
    0x7a, 0x4c, 0x5e, 0xf0, 0x3e, 0x0d, 0xb9, 0x40,
    0x4e, 0x4e, 0xee, 0xb3, 0x5d, 0xdf, 0x15, 0x1d,
    0x23, 0x9e, 0x8b, 0x78, 0xc2, 0x64, 0x08, 0x24,
    0xce, 0x1f, 0x10, 0x6e, 0xab, 0x1c, 0x01, 0x9a,
    0xca, 0xd3, 0x98, 0x56, 0x31, 0xc7, 0x0c, 0x36,
```

```
    0x3f, 0x30, 0x15, 0xf5, 0xec, 0x41, 0xc8, 0x82,
    0x5e, 0xc4, 0xf4, 0x7f, 0x9e, 0xa0, 0x4d, 0x7e,
    0xdc, 0x17, 0x34, 0x1f, 0x5c, 0x41, 0x98, 0x9c,
    0x56, 0x3c, 0x6a, 0xc2, 0xac, 0x4e, 0xd8, 0xac,
    0x6b, 0xa4, 0x61, 0xfc, 0xaf, 0xb0, 0xb4, 0x1e,
    0x64, 0x4b, 0x00, 0x3c, 0xa3, 0xcf, 0x52, 0x60,
    0x73, 0xa1, 0xef, 0x97, 0x21, 0x7d, 0xf0, 0x3e,
    0x26, 0xbb, 0xd0, 0x22, 0xee, 0x27, 0x9f, 0x06,
    0x95, 0x3c, 0xa3, 0xcd, 0xfd, 0xb4, 0x3d, 0x49,
    0x20, 0xf3, 0x2e, 0xd6, 0x87, 0xd7, 0x81, 0x11,
    0x32, 0x84, 0xb1, 0x7d, 0x34, 0x10, 0x72, 0x58,
    0x1a, 0x3b, 0x38, 0xe7, 0x9f, 0x65, 0xd7, 0x54,
    0x9f, 0x80, 0x39, 0x00, 0x74, 0x5f, 0x37, 0x94,
    0xbf, 0x71, 0x75, 0xa8, 0xca, 0xeb, 0x62, 0xb7,
    0x96, 0x6f, 0xf7, 0xa2, 0xb7, 0x0f, 0xdf, 0x1f,
    0x12, 0x3f, 0x98, 0x26, 0x65, 0x2e, 0xda, 0x09,
    0x7e, 0x7f, 0x39, 0x2d, 0xf8, 0xd0, 0xa9, 0xc4,
    0xf4, 0x4b, 0xa4, 0x0e, 0x54, 0xb9, 0x71, 0xbe,
    0x31, 0x87, 0x6f, 0x1e, 0x43, 0xaa, 0x1f, 0x65,
    0xf5, 0xa6, 0x0e, 0xbf, 0x53, 0xf1, 0xea, 0x9b,
    0x8f, 0x9b, 0xc6, 0x37, 0x31, 0xfa, 0xbb, 0xb4,
    0xdf, 0xcb, 0xd2, 0xbc, 0xa9, 0x94, 0x70, 0x37,
    0x8f, 0x5a, 0x91, 0xc2, 0xf1, 0xbc, 0xb0, 0x80,
    0x10, 0xea, 0xfa, 0x3e, 0x32, 0xf3, 0xac, 0xe6,
    0xd3, 0xc9, 0xe9, 0x1d, 0x12, 0xd7, 0x9a, 0x78,
    0x3d, 0xb3, 0xf8, 0xdf, 0xec, 0xdd, 0xd8, 0x1a,
    0xda, 0xb8, 0x79, 0x03, 0x75, 0x28, 0x8c, 0x5d,
    0xf9, 0xee, 0xa4, 0xa6, 0x63, 0xb5, 0x45, 0x6a,
    0x02, 0xdc, 0x4f, 0xe4, 0x4c, 0xd9, 0x82, 0x1c,
    0x77, 0x3b, 0xdc, 0xfd, 0xf8, 0xc5, 0xe0, 0x68,
    0x65, 0x22, 0xab, 0x40, 0x98, 0x50, 0x01, 0x0f,
    0x34, 0xe9, 0x0a, 0x64, 0x2c, 0x0a, 0x96, 0xf2,
    0xbd, 0xa3, 0xe9, 0x75, 0x8b, 0xfd, 0xd5, 0x18,
    0x47, 0xa7, 0x15, 0xb0, 0xb8, 0xcf, 0x12, 0xc2,
    0x29, 0xf4, 0x39, 0x3d, 0xa6, 0xc8, 0x49, 0x72,
    0xf7, 0x3f, 0x2b, 0x2f, 0x72, 0xb7, 0x5d, 0x03,
    0x23, 0xe5, 0x9a, 0x48, 0xe3, 0xf2, 0x08, 0xe6,
    0x6d, 0xe7, 0x2f, 0x4d, 0x9a, 0x44, 0x04, 0x75,
    0x2a, 0xc7, 0x0f, 0x04, 0xe6, 0x47, 0x25, 0x27,
    0x1b, 0xd3, 0xff, 0xf2, 0x6c, 0xd7, 0xb4, 0x19,
    0x1d, 0x0d, 0xe3, 0xf7, 0x19, 0x63, 0xd7, 0x6e,
    0xf5, 0xda, 0x72, 0xbf, 0x7e, 0xf6, 0xd4, 0xdb,
    0xd7, 0x87, 0xce, 0xa1, 0x8a, 0x13, 0x6f, 0x01,
    0x2b, 0x2d, 0x8c, 0x8b, 0x50, 0x83, 0xdd, 0xcc,
    0xf8, 0xc2, 0x86, 0x41, 0xb6, 0x25, 0x60, 0x17,
    0x5f, 0x6d, 0x28, 0xea, 0xdd, 0xa5, 0xc9, 0xa1,
    0x5b, 0xf1, 0x53, 0xa5, 0xfd, 0x01, 0x16, 0xdf,
    0xd4, 0xf5, 0x62, 0x2a, 0x8f, 0x18, 0xd0, 0x7d,
    0x55, 0x93, 0x03, 0xe2, 0xe8, 0xdd, 0x10, 0x1c,
    0x17, 0x0f, 0xe8, 0x35, 0x88, 0xfb, 0xe2, 0x00,
    0x5e, 0x90, 0x07, 0x1b, 0xb0, 0x70, 0x64, 0xcd,
    0x36, 0x2e, 0x15, 0x32, 0x31, 0x1c, 0x06, 0x7e,
    0xf4, 0xa7, 0xa5, 0x00, 0xe3, 0x5e, 0x20, 0xc5,
    0x82, 0x05, 0x98, 0x18, 0xb3, 0x3e, 0xd0, 0x66,
    0x3f, 0x7a, 0xe0, 0xa0, 0xb2, 0xc8, 0x87, 0xef,
    0x72, 0x30, 0x91, 0x79, 0x9f, 0xaf, 0xfd, 0xbb,
},
```

```
  &case4
};


test_case_t case2 = {
  /* key */
  {
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f,
  },
  /* bytes in key */
  16,
  /* plaintext */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
  },
  /* bytes in plaintext */
  48,
  /* associated data */
  { 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x97, 0xc6, 0xb2, 0xb7, 0x19, 0xa9, 0x54, 0xe3,
    0x3b, 0xab, 0x39, 0x0a, 0xf2, 0x57, 0xeb, 0x4c,
    0x59, 0x93, 0xdd, 0x9a, 0x1a, 0x36, 0x61, 0xd5,
    0xb1, 0x52, 0xf8, 0xd6, 0x5f, 0x35, 0x37, 0xb9,
    0x54, 0x34, 0xff, 0xf3, 0x35, 0x2d, 0xfe, 0xb6,
    0x61, 0x5e, 0xc1, 0xb1, 0xc6, 0x6d, 0x81, 0x5d,
  },
  &case3
};

test_case_t case1 = {
  /* key */
  {
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f,
    0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17,
    0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f,
  },
  /* bytes in key */
  32,
```

```
  /* plaintext */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
  },
  /* bytes in plaintext */
  32,
  /* associated data */
  { 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0x0a, 0xa2, 0x7c, 0x16, 0x7b, 0x7a, 0x6f, 0x13,
    0x93, 0x23, 0x4c, 0xb1, 0x82, 0x8f, 0x73, 0x7c,
    0xe5, 0x3d, 0xa9, 0xf5, 0x05, 0x8e, 0xbd, 0x81,
    0xf4, 0x4b, 0xfb, 0x8a, 0xa6, 0x4a, 0xe6, 0xc1,
  },
  &case2
};

test_case_t case0 = {
  /* key */
  {
    0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
    0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f,
  },
  /* bytes in key */
  16,
  /* plaintext */
  {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in plaintext */
  32,
  /* associated data */
  {
    0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  },
  /* bytes in associated data */
  16,
  /* ciphertext */
  {
    0xf7, 0x27, 0xd7, 0x48, 0xb8, 0x6e, 0x3b, 0x36,
    0x2f, 0x20, 0x81, 0x0e, 0xed, 0xbe, 0x37, 0x8a,
```

```
    0x07, 0x76, 0x16, 0x31, 0xb9, 0x00, 0x94, 0x54,
    0xd5, 0x4d, 0x8d, 0x94, 0x9c, 0x35, 0x27, 0x19,
  },
  &case1
};
```