Recitation 6, Authenticated Encryption

Kevin King (kcking@mit.edu)

Administrivia

- Form your problem set 3 groups!

Outline

- One-Time MACs
- Composition of Encryption and Authentication
  - Security Definitions
  - Analyses
    - http://cseweb.ucsd.edu/~mihir/papers/oem.pdf

One-Time MACs

More details at: http://web.mit.edu/6.857/OldStuff/Fall97/lectures/lecture3.pdf

What is wrong with One-Time Pad?

- Completely malleable: flipping a bit of the ciphertext flips the same bit of the plaintext

How can we fix this?

- require a tag along with message that satisfies some equation

Idea 1

- work in $\mathbb{Z}_p^*$
- draw random $a \leftarrow \mathbb{Z}_p$
- $\mathsf{Tag}_a(m) = m^{-1}a \bmod p$
- $\mathsf{Verify}_a(m, t)$:
  - If $m * t == a$:
    - output ACCEPT
  - Else:
    - output $\perp$
- Does this work. . . ?

  - No! It is malleable!
  - Multiply m by x and t by x^{-1}, and equation still holds
- Only one slightly redeeming quality... need one message to be sent in order to forge
- How can we extend this?

Idea 2

- Still work in $\mathbb{Z}_p$
- This time, utilize both $*$ and $+$
- draw random $a, b \leftarrow \mathbb{Z}_p$
- $\mathsf{Tag}_{a,b}(m) = a * m + b \bmod p$
- $\mathsf{Verify}_{a,b}(m, t)$:
  - If $a * m + b == t \bmod p$:
    - output ACCEPT
  - Else:
    - output $\perp$
- It works!
- $a * m + b \bmod p$ is effectively a line
  - (m, t) provides a single point on the line, but p many different lines (or a, b combinations) run through this point
- What happens when we reuse a key?
  - Eve now has two equations with two unknowns a, b, she can solve for them!

Composition of Encryption and Authentication

The remainder of this lecture is a summary of http://cseweb.ucsd.edu/~mihir/papers/oem.pdf

Security Definitions

The first step to establishing any form of security is to determine the adversarial model. How much power do we want to give the adversary? What don't we want them to be able to do with that power?

Once we have these definitions, we can go about creating schemes that meet them or analyzing schemes that do not. Today we will use the definitions to analyze composition of schemes that provide authentication and confidentiality, but these same techniques can be applied to any situation where we want to show that given our assumptions, a scheme meets a certain defintion of security (for example, factoring being difficult implying the security of RSA).

Indistinguishability of Ciphertexts

> Confidentiality properties for encryption schemes

Why indistinguishability of ciphertexts? One could imagine other security definitions such as (1) the key is not revealed, or (2) the adversary cannot decrypt any messages. Even if (1) holds true, there is no guarantee that the adversary doesn't learn the plaintext. Similarly for (2), even if the adversary does not learn the entire message, they could learn everything but one bit and this definition of security would still be satisfied.

Indistinguishability of ciphertexts implies that the adversary gains negligible information about which message a ciphertext would decrypt to. This notion implies both hiding of the secret key and the adversary being unable to decrypt a ciphertext, making it a stronger definition.

IND-CPA

- Indistinguishability under chosen plaintext attack
- Adversary chooses many messages and receives corresponding message-ciphertext pairs
- Adversary then chooses two challenge messages (m_0, m_1)
- IND-CPA $\implies$ adversary, given $Enc(m_{challenge})$ cannot guess the encrypted message with more than negligible advantage.
- In practice: You (the adversary) can control what GMail encrypts by sending a message to your friend and listening to the network traffic when your friend checks their email.
- Requires a non-deterministic encryption function

IND-CCA

- Indistinguishability under chosen ciphertext attack
- Adversary chooses many messages and many ciphertexts, receives corresponding message-ciphertext pairs
- Adversary then chooses two challenge messages (m_0, m_1)
- IND-CCA $\implies$ adversary, given $Enc(m_{challenge})$, cannot guess the encrypted message with more than negligible advantage.
- IND-CCA is basically the maximum amount of power we can give to the adversary without revealing the secret key. We equate it in practice to a "lunchtime attack," where the adversary sneaks into your office during your lunch break and has full access to the decryption circuit (but luckily the key is kept in some secure hardware component).
- IND-CCA provides non-malleability, and malleability is how we broke the IND-CPA block cipher schemes

Unforgeability

Properties for pure authentication schemes

WUF-CMA

- Weak unforgeability under chosen message attack
- Adversary receives tagging oracle
- WUF-CMA $\implies$ adversary cannot create a valid tag for a new message.

SUF-CMA

- Strong unforgeability under chosen message attack
- Adversary receives tagging oracle
- SUF-CMA $\implies$ adversary cannot create a new valid (m', t') pair, even for a message already queried on the tagging oracle

Integrity

Authentication properties for authenticated encryption schemes

INT-PTXT

Note: in symmetric schemes we call the signature a `tag` because there is no identity associated with it.

- Integrity of plaintext
- Adversary chooses many messages m_i and receives corresponding message-tag pairs (m_i, t_i)
- INT-PTXT $\implies$ adversary cannot construct a pair (m', t') for a new m' such that $\mathsf{Verify}((m', t'))$ outputs ACCEPT
- In practice: You (the adversary) can control what GMail tags by sending messages to yourself.
- Can be deterministic

INT-CTXT

- Integrity of ciphertext (includes encryption and authentication)
- Adversary chooses many messages m_i and receives corresponding ciphertexts c_i
- INT-CTXT $\implies$ adversary cannot construct a new ciphertext c' that successfully decrypts to any message

Relationships Between Security Definitions

- INT-PTXT $\not\implies$ INT-CTXT
  - One way to prove a separation of two definitions is to adversarially construct a scheme that violates one but not the other.
  - We will take an INT-PTXT scheme $(\mathcal{T}, \mathcal{V})$ and modify it, preserving INT-PTXT but trivially breaking INT-CTXT
  - $\mathcal{T}'(m)$:
    - output $t = 0||\mathcal{T}(m)$
  - $\mathcal{V}'(t, m)$:
    - let $t' = b||t$
    - output $\mathcal{V}(t)$
  - $(\mathcal{T}', \mathcal{V}')$ is still INT-PTXT because we must forge a tag to a new m' and the 0 bit will not help.
  - Adversary A breaks INT-CTXT by flipping first bit of tag to 1
- INT-CTXT $\implies$ INT-PTXT
  - A common technique to prove implication is to show that we can create an adversary for the left side of the implication assuming an adversary for the right side (equivalent to the contrapositive).
  - Prove contrapositive ~INT-PTXT $\implies$ ~INT-CTXT
  - INT-PTXT adversary A can create a ciphertext c' that decrypts to a new message m'. Since decryption is deterministic and unique, this must be a new ciphertext
  - $\therefore$ c' breaks INT-CTXT
- INT-CTXT $\wedge$ IND-CPA $\implies$ IND-CCA
  - Prove contrapositive ~IND-CCA $\implies$ ~(INT-CTXT $\wedge$ IND-CPA)
  - Rewrite... ~IND-CCA $\implies$ ~INT-CTXT $\vee$ ~IND-CPA
  - In words, distinguishability of ciphertexts either breaks integrity of ciphertexts or indistinguishability of plaintexts.
  - Start with IND-CCA adversary $\mathcal{A}$, but we no longer have a decryption oracle.
  - Two cases:
    - $\mathcal{A}$ used the decryption oracle to break IND-CCA

· Then $\mathcal{A}$ queried the decryption oracle with some new c' that successfully decrypted to some m', and c' breakts INT-CTXT

* $\mathcal{A}$ did not use the decryption oracle, in which case we can simulate an IND-CPA game and preserve $\mathcal{A}$'s advantage

Encrypt-and-MAC

- Algorithm:
  - Encryption key k_e, Authentication key k_a, Message m
  - $\mathsf{AuthEnc}_{k_e||k_a}(m)$:
    * $c \leftarrow \mathsf{Enc}_{k_e}(m)$
    * $t \leftarrow \mathsf{Tag}_{k_a}(m)$
    * output (c, t)
  - $\mathsf{AuthDec}_{k_e||k_a}((c, t))$:
    * $m' \leftarrow \mathsf{Dec}_{k_e}(c)$
    * $t' \leftarrow \mathsf{Verify}_{k_a}((m', t))$
    * If $t' == \mathsf{ACCEPT}$:
      · output m'
    * Else:
      · output $\perp$
- Analysis:
  - INT-PTXT:
    * satisfied directly by MAC
  - IND-CPA:
    * for simplicity, assume we are using a deterministic MAC
      · then MAC reveals which message was encrypted again during challenge phase
      · !! not even IND-CPA !!
      · (still not IND-CPA with randomized MAC, see paper for details)

MAC-then-Encrypt

- Algorithm:
  - $\mathsf{AuthEnc}_{k_e||k_a}(m)$:
    * output $c = \mathsf{Enc}_{k_e}(m||\mathsf{Auth}_{k_a}(m))$
  - $\mathsf{AuthDec}_{k_e||k_a}(c)$:
    * $m'||t' \leftarrow \mathsf{Dec}_{k_e}(c)$
    * If $\mathsf{Verify}_{k_a}(m', t') == \mathsf{ACCEPT}$:
      · output m'
    * Else:
      · output $\perp$
- Analysis:
  - INT-PTXT
    * given INT-PTXT adversary $\mathcal{I}$ for $\mathsf{AuthEnc}$, we can construct adversary $\mathcal{F}$ to break INT-PTXT of Auth

            - $\mathcal{F}$ draws an encryption key k_e
            - provides $\mathsf{AuthEnc}$ oracle to $\mathcal{I}$ using own Enc_{k_e} and MAC Auth oracle
            - given response ciphertext c' from $\mathcal{F}$, decrypt and forward (m', t') to MAC Verify challenger
            - $\therefore$ INT-PTXT retained from MAC
    - IND-CPA
        * given IND-CPA adversary $\mathcal{A}$ for $\mathsf{AuthEnc}$, we can construct an adversary $\mathcal{P}$ to break IND-CPA of Enc
            - $\mathcal{P}$ draws an authentication key k_a
            - provides $\mathsf{AuthEnc}$ oracle to $\mathcal{A}$ by running Tag_{k_a} and forwarding to Enc oracle
            - output $\mathcal{A}$'s answer to the Enc challenger

Encrypt-then-MAC

- Algorithm:
    - $\mathsf{AuthEnc}_{k_e||k_a}(m)$:
        * $c \leftarrow \mathsf{Enc}_{k_e}(m)$
        * $t \leftarrow \mathsf{Auth}_{k_a}(c)$
        * output $c||t$
    - $\mathsf{AuthDec}_{k_e||k_a}(c||t)$:
        * If $\mathsf{Verify}_{k_a}(c, t) == \mathsf{ACCEPT}$:
            - output $\mathsf{Dec}_{k_e}(c)$
        * Else:
            - output $\perp$
- Analysis:
    - Results differ depending on strength of MAC
    - Assuming WUF-CMA:
        * same guarantees as MAC-then-Encrypt (INT-PTXT, IND-CPA)
        * proof similar to MAC-then-Encrypt, recreate full challenge by emulating either encryption or authentication
    - Assuming SUF-CMA:
        * INT-CTXT
            - given INT-CTXT adversary $\mathcal{I}$ for $\mathsf{AuthEnc}$, we can construct an adversary $\mathcal{F}$ to break SUF-CMA of Auth
            - $\mathcal{F}$ draws encryption key k_e
            - answer $\mathsf{AuthEnc}(m)$ oracle queries with $\mathsf{Enc}_{k_e}(m)$, then pass to Auth oracle
            - answer $\mathsf{AuthDec}(c)$ oracle queries with $\mathsf{Dec}_{k_e}(c)$, pass to Verify oracle
            - decrypt challenge response from $\mathcal{I}$ of $c' = (m', t')$
            - c' must be a new ciphertext that passes by definition of $\mathcal{I}$, which means (m', t') is also new and SUF-CMA of MAC has been broken
        * $\implies$ IND-CCA security!

Why do we care?

- TLDR: Encrypt-then-MAC most secure given SUF-CMA authentication
- IND-CCA is the strongest reasonable notion of security we have
- Authenticated Encryption schemes now provide CCA without any composition, use these instead (EAX, GCM)
- Lots of scheme composition in the wild still, good to know how to analyze it
- Never roll your own crypto!