INT-RUP Analysis of Block-cipher Based Authenticated Encryption Schemes

Avik Chakraborti, Nilanjan Datta and Mridul Nandi

Indian Statistical Institute, Kolkata

CTRSA-2016, San Francisco, USA

Outline of the talk

1. Introduction.
2. Our Contribution.
3. Conclusion

1 Introduction

2 Our Contribution

3 Conclusions

Authenticated Encryption (AE)

Why AE?

- Privacy of Plaintext.
- Authenticity of the plaintext/ ciphertext and associated data.

More Formally....

- Tagged-encryption : AE.enc : $\mathcal{M} \times \mathcal{D} \times \mathcal{N} \times \mathcal{K} \rightarrow (\mathcal{C} \times \mathcal{T})$
- Verified-decryption : AE.dec : $(\mathcal{C} \times \mathcal{T}) \times \mathcal{D} \times \mathcal{N} \times \mathcal{K} \rightarrow \mathcal{M} \cup \perp$

Intigrity Security AE

Intigrity Security of AE

- Integrity Security of AE when adversary is given Encryption and Verification oracle.

- Encryption Query: $(N_i, D_i, M_i) \rightarrow (C_i, T_i)$
Verificationtion Query: $(N_i, D_i, (C_i, T_i)) \rightarrow M_i / \perp$

- $\mathbf{Adv}_{\pi}^{int-ctxt}(A) = |Pr[K \in_R \mathcal{K} : A^{\mathcal{E}_K, \mathcal{V}_K} \neq \perp]|$

INT-RUP Security and rate of Block Cipher based AE

INT-RUP Security of AE Construction (Andreeva et.al.)

- Adversary is given both Encryption, Decryption and Verification oracle.
- Decryption Query: $(N_i, D_i, C_i) \rightarrow M_i$ (no T_i in the query)
- $\mathbf{Adv}_{\pi}^{int-rup}(A) = |Pr[K \in_R \mathcal{K} : A^{\mathcal{E}_K, \mathcal{D}_K, \mathcal{V}_K} \neq \perp]|$
- Used for low-end devices with limited buffer.

Rate of a Construction

- Messages blocks processed per block-cipher call.
- Rate-1 means efficient construction.

Affine Mode AE

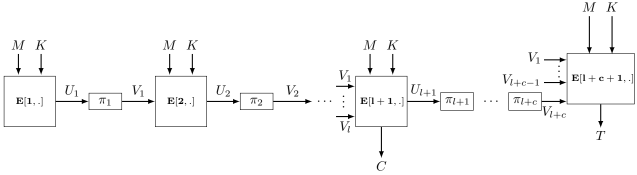


Figure : Structure of Affine Mode AE Schemes

Affine Mode AE - Encryption

Matrix Representation

$$E.\begin{pmatrix} L \\ M \\ Y^* = \begin{pmatrix} Y \\ Y_{tag} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} X^* = \begin{pmatrix} X \\ X_{tag} \end{pmatrix} \\ Z = \begin{pmatrix} C \\ T \end{pmatrix} \end{pmatrix}$$

Encryption Matrix Representation

- E : encryption matrix, L : key vector, M is message vector
- Y : Intermediate output from π during M Processing
- Y_{tag} : Intermediate output from π during *tag* Processing
- X : Intermediate input to π during M Processing
- X_{tag} Intermediate input to π during *tag* Processing
- C : ciphertext vector, T : *tag* vector

1 Introduction

2 Our Contribution

- INT-RUP Insecurity of Affine mode AE
- INT-RUP Insecurity of CPFB
- mCPFB: A rate $\frac{3}{4}$ INT-RUP secure construction

3 Conclusions

Our Contribution

Result 1.

rate-1 Affine mode Authenticated Encryption mode is INT-RUP insecure.

Significance of the Result

Guideline: To achieve INT-RUP security, one has to compromise efficiency.

Our Contribution

Result 2.

CPFB (rate $\frac{3}{4}$) is INT-RUP insecure.

Questions

- How much efficiency we have to loose to get INT-RUP security?
- Can we have an INT-RUP secure scheme with rate $\frac{3}{4}$?

Our Contribution

Result 3.

m-CPFB (rate $\frac{3}{4}$) is INT-RUP insecure.

Significance

- INT-RUP comes with small degrade in efficiency.
- “rate-1” - a borderline criteria for INT-RUP security.

INT-RUP Attack

Queries of INT-RUP Adversary

- Encryption Query: $(N, AD, M^0 = (M_1^0, M_2^0, \ldots, M_l^0))$. Let, $C^0 = (C_1^0, C_2^0, \ldots, C_l^0, T^0)$ be the tagged ciphertext.
- Unverified Plaintext Query: $(N, AD, C^1 = (C_1^1, C_2^1, \ldots, C_l^1))$. Let $M^1 = (M_1^1, M_2^1, \ldots, M_l^1)$ be the corresponding plaintext.
- Forged Query: $(N, AD, C^f = (C_1^f, C_2^f, \ldots, C_l^f), T^f)$, which realizes a $\delta = (\delta_1, \ldots, \delta_l)$ sequence.

C^f realizes a $\delta = (\delta_1, \ldots, \delta_l)$-sequence

$\forall i \leq l,\ U_i^f = U_i^{\delta_i}$ and $\forall i > l,\ U_i^f = U_i^0$.

Structure of Decryption Matrix

During Message Process

- Observed by Enc and Dec queries
- $\Delta C^{ij}, \Delta V^{ij} \Delta U^{ij}$ and ΔM^{ij} are observed differences.

$$\begin{pmatrix} D_{12} & D_{13} \\ D_{32} & D_{33} \end{pmatrix} \cdot \begin{pmatrix} \Delta C^{ij} \\ \Delta V^{ij} \end{pmatrix} = \begin{pmatrix} \Delta U^{ij} \\ \Delta M^{ij} \end{pmatrix}, \quad i = 0, j \in \{1, f\}$$

During Tag Process

- $\Delta C^{0f}, \Delta V^{0f}, \Delta V^{0f}_{tag} U^{0f}_{tag}$, and ΔT^{0f} are observed differences.

$$\begin{pmatrix} D_{22} & D_{23} & D_{24} \\ D_{42} & D_{43} & D_{44} \end{pmatrix} \cdot \begin{pmatrix} \Delta C^{0f} \\ \Delta V^{0f} \\ \Delta V^{0f}_{tag} \end{pmatrix} = \begin{pmatrix} \Delta U^{0f}_{tag} \\ \Delta T^{0f} \end{pmatrix}$$

INT-RUP Attack (Construction of Forged Query)

Step I: Find ΔV^{01}

$$\Delta V^{01} = D_{33}^{-1}(\Delta M^{01} + D_{32}\Delta C^{01})$$

Step II: Find ΔC^{0f} in terms of δ

$$\begin{aligned} \Delta C^{0f} &= D_{12}^{-1}.(\Delta U^{0f} + D_{32}\Delta V^{0f}) \\ &= D_{12}^{-1}(\delta.U^{01} + D_{32}\delta.V^{01}) \\ &= D^{*}.\delta \end{aligned}$$

INT-RUP Attack (Construction of Forged Query)

Step III: Find δ that makes $\Delta U^{0f}_{tag} = 0$

Solve the following set of equations to find a δ:

$$D_{22}\Delta C^{0f} + D_{23}\Delta V^{0f} \quad = \quad 0$$

This equation has at least one solution as long as $l > (c-1).n$

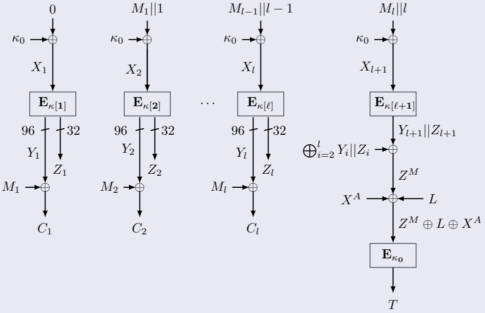
Step IV: Find ΔC^{0f} and ΔT^{0f}

Put $\delta = \delta^*$ in the following equations:

$$\begin{aligned} \Delta C^{0f} &= D_{12}^{-1}.D^*.\delta \\ \Delta T^{0f} &= D_{42}\Delta C_{0f} + D_{43}\Delta V_{0f} \end{aligned}$$

Revisit CPFB

Encryption and Tag Genration of CPFB



INT-RUP Attack on CPFB

INT-RUP Attack on CPFB

1. Encryption query: (N, A, M^0), $|M^0| = l = 129$. Let C^0 be the ciphertext
2. Unverified Plaintext decryption query: (N, A, C^1) of length l. Let, M^1 be the corresponding plaintext.
3. Compute Y values: $Y_1^0, \cdots, Y_l^0$ and $Y_1^1, \cdots, Y_l^1$ from the two queries (by $M^0 + C^0$ and $M^1 + C^1$).
4. Find the δ-sequence: $\delta = (\delta_1, \ldots, \delta_l)$, with $\delta_1 = 0$ such that,
$$\textstyle\sum_{i=2}^{l}(Y_i^{\delta_i}||Z_i^{\delta_i}) = \sum_{i=2}^{l}(Y_i^0\ ||Z_i^0).$$
Expect 2^{32}-many such δ-sequences.

INT-RUP Attack on CPFB

INT-RUP Attack on CPFB

Perform the following for all such δ-sequence:

1. Set $C_1^f = C_1^0$. For all $1 < i < l$, set $C_i^f = C_i^{\delta_i}$ if $\delta_{i-1} = \delta_i$ and $C_i^{\delta_i} + Y_i^0 + Y_i^1$, otherwise.
2. Set $C_l^f = C_l^0$ if $\delta_l = 0$. Else, set $C_l^f = C_l^0 + Y_l^0 + Y_l^1$.
3. Return $(C_1^f, C_2^f, \cdots, C_l^f, T^0)$ as forged Ciphertext.

Building an INT-RUP Secure rate-$\frac{3}{4}$ Construction

Potential Weakness of CPFB

1. Y_i values can be observed. Only Z_i-values are unknown.
2. Z_i has only 32-bit entropy on the Tag.

Requirement of the New Construction

- Ensure 128-bit entropy of Z-values on the tag.
- Ensure at-least 4 different Z-values for 2 messages of same length.

mCPFB: modified CPFB

Introduce ECC Code

Expand $M = (M_1, \ldots, M_l)$ by a Distance 4 Error Correcting Code ECCode :

$$\mathrm{ECCode}(M) = (M_1, \ldots, M_l, M_{l+1}, M_{l+2}, M_{l+3})$$
$$(M_{l+1}, M_{l+2}, M_{l+3}) = V_\beta^{(3,l)} \cdot M$$

Produce 128-bit entropy of Z-values during Tag Generation:

Update Z^M as follows:

$Z_M = V_\alpha^{(4,l+3)} \cdot (Z_2, Z_3, \cdots, Z_{l+3}, Z_{l+4}) \oplus (0^{32} || (Y_2 \oplus \cdots \oplus Y_{l+3}))$

mCPFB: modified CPFB

Changes in the keys

- κ_0 is used as the masking key only.
- κ_1 is used as block-cipher key for AD processing.
- $\kappa_1, \ldots, \kappa_{-2}$ is used as block-cipher keys for message processing.
- κ_{-1} is used as block-cipher key for tag and producing L-values.

INT-RUP Security of mCPFB

Claim 1

Consider the function f that takes N, I and i as input and outputs O such that $O = E_{\kappa[i]}(I||(i \ mod \ 2^{32}) + \kappa_0)$ where $\kappa[i] = E_K(N||j||I)$, $j = \lceil \frac{i}{2^{32}} \rceil$. f is assumed to have (q, ϵ)-PRF security where ϵ is believed to achieve beyond birthday security.

INT-RUP advantage

f: $(q_e + q_r, \epsilon)$-PRF. Any adversary $\mathcal{A}$ with q_e many encryption query and q_r many unverified plaintext queries, one forgery attempts, has the advantage:

$$Adv^{int\_rup}_{mCPFB}(\mathcal{A}) \leq \frac{5}{2^{128}} + \epsilon$$

Proof Sketch

Argument for Different Cases

- (Case A) $\forall i, N^* \neq N_i$: Through randomness of κ_{-1}.
- (Case B) $\exists$ unique $i \ni N^* = N_i, T^* \neq T_i$: Through randomness of κ_{-1}.
- (Case C) $\exists$ unique $i \ni N^* = N_i,\ T^* = T_i,\ |C_i| = |C^*|$: Through randomness of Z_i's.
- (Case D) $\exists$ unique $i \ni N^* = N_i,\ T^* = T_i,\ |C_i| \neq |C^*|$: Through randomness of κ_{-1}.

1. Introduction
2. Our Contribution
3. Conclusions

Conclusions

- INT-RUP attack on any “Rate-1” affine AE mode

- INT-RUP attack on a “Rate-$\frac{3}{4}$” AE scheme *CPFB*

- Proposal of *mCPFB* : an INT-RUP secure scheme

Thank you

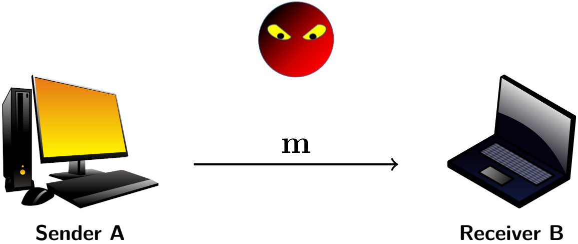
From Stateless to Stateful: Generic Authentication and Authenticated Encryption Constructions with Application to TLS

Colin Boyd[1] ***Britta Hale***[1]
Stig Frode Mjølsnes[1] *Douglas Stebila*[2]

[1]Norwegian University of Science and Technology
[2]Queensland University of Technology

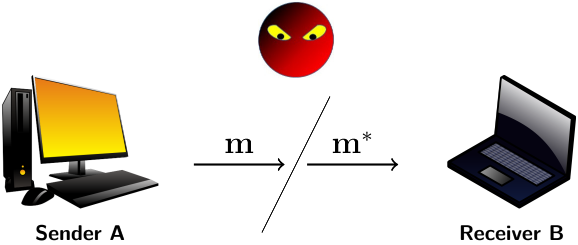
1 March 2016

What *is* data authentication?



Is $\mathbf{m}$ from **A**?

What *is* data authentication?

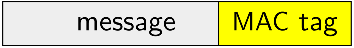


Is $\mathbf{m}$ from **A**?

Has $\mathbf{m}$ been modified?

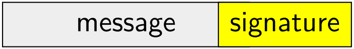
ACHIEVING AUTHENTICATION

- Message Authentication Code (MAC)
  - HMAC, etc...



message MAC tag

- Signatures
  - DSA, Elliptic Curve DSA, etc...

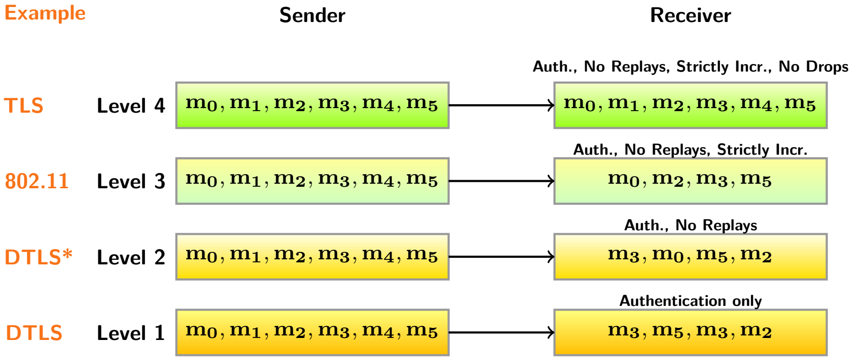


message signature

- Authenticated Encryption with Associated Data (AEAD)
  - Galois Counter Mode (GCM), etc...

ciphertext

AUTHENTICATION HIERARCHY



Example		Sender	Receiver
TLS	Level 4	$m_0, m_1, m_2, m_3, m_4, m_5$	Auth., No Replays, Strictly Incr., No Drops $m_0, m_1, m_2, m_3, m_4, m_5$
802.11	Level 3	$m_0, m_1, m_2, m_3, m_4, m_5$	Auth., No Replays, Strictly Incr. m_0, m_2, m_3, m_5
DTLS*	Level 2	$m_0, m_1, m_2, m_3, m_4, m_5$	Auth., No Replays m_3, m_0, m_5, m_2
DTLS	Level 1	$m_0, m_1, m_2, m_3, m_4, m_5$	Authentication only m_3, m_5, m_3, m_2

SECURE CHANNEL VARIATIONS

Level 1
Canetti–Krawczyk 2001
Generic network channel
protocol description

Level 4
Jager–Kohlar–Schäge–
Schwenk 2012

Level 4
Bellare–Kohno–Namprempre 2002
INT-SFCTXT from INT-CTXT

Level 4
Krawczyk–Paterson–Wee 2013

Level 4
Canetti–Krawczyk 2001
Network authentication protocol

Level 1
Paterson–
Ristenpart–
Shrimpton 2011

HIERARCHY OF AUTHENTICATION

$\mathsf{Exp}^{\mathsf{auth}_i}_{\Pi,\mathcal{A}}()$:

```
$k \xleftarrow{\$} \mathrm{Kgn}()$
$st_{\mathrm{E}} \leftarrow \perp, st_{\mathrm{D}} \leftarrow \perp$
$u \leftarrow 0,\ v \leftarrow 0$
$\mathtt{r} \leftarrow 0$
$\mathcal{A}^{\mathsf{Send}(\cdot),\mathsf{Recv}(\cdot)}()$
return $\mathtt{r}$
```

Oracle Send(m):

```
$u \leftarrow u + 1$
$(sent_u, st_{\mathrm{E}}) \leftarrow \mathrm{Snd}(k, m, st_{\mathrm{E}})$
return $sent_u$ to $\mathcal{A}$
```

Oracle Recv(c):

```
$v \leftarrow v + 1$
$rcvd_v \leftarrow c$
$(m, \alpha, st_{\mathrm{D}}) \leftarrow \mathrm{Rcv}(k, c, st_{\mathrm{D}})$
if $(\alpha = 1) \wedge \boxed{\mathsf{cond}_i}$ then
    $\mathtt{r} \leftarrow 1$
    return $\mathtt{r}$ to $\mathcal{A}$
end if
return $\perp$
```

1. **Basic authentication:**
   $\mathsf{cond}_1 = (\nexists w : c = sent_w)$
2. **Basic authentication, no replays:**
   $\mathsf{cond}_2 = (\nexists w : c = sent_w) \vee (\exists w < v : c = rcvd_w)$
3. **Basic authentication, no replays, strictly increasing:**
   $\mathsf{cond}_3 = (\nexists w : c = sent_w) \vee (\exists w, x, y : (w < v) \wedge (sent_x = rcvd_w) \wedge (sent_y = rcvd_v) \wedge (x \geq y))$
4. **Basic authentication, no replays, strictly increasing, no drops:**
   $\mathsf{cond}_4 = (u < v) \vee (c \neq sent_v)$

HIERARCHY OF AEAD

```
$\mathsf{Exp}^{\mathsf{aead}_i - b}_{\Pi,\mathcal{A}}()$:
  $k \xleftarrow{\$} \mathrm{Kgn}()$
  $st_{\mathrm{E}} \leftarrow \perp, st_{\mathrm{D}} \leftarrow \perp$
  $u \leftarrow 0,\ v \leftarrow 0$
  $\mathsf{phase} \leftarrow 0$
  $b' \xleftarrow{\$} \mathcal{A}^{\mathsf{Encrypt}(\cdot),\mathsf{Decrypt}(\cdot)}()$
  return $b'$
```

```
Oracle $\mathsf{Encrypt}(l, \mathsf{ad}, m_0, m_1)$:
  $u \leftarrow u + 1$
  $(sent.c^{(0)}, st_{\mathrm{E}}^{(0)}) \leftarrow \mathrm{E}(k, l, \mathsf{ad}, m_0, st_{\mathrm{E}})$
  $(sent.c^{(1)}, st_{\mathrm{E}}^{(1)}) \leftarrow \mathrm{E}(k, l, \mathsf{ad}, m_1, st_{\mathrm{E}})$
  if $sent.c^{(0)} = \perp$ or $sent.c^{(1)} = \perp$ then
      return $\perp$
  end if
  $(sent.ad_u, sent.c_u, st_{\mathrm{E}}) := (\mathsf{ad}, sent.c^{(b)}, st_{\mathrm{E}}^{(b)})$
  return $sent.c_u$
```

```
Oracle $\mathsf{Decrypt}(\mathsf{ad}, c)$:
  if $b = 0$ then
      return $\perp$
  end if
  $v \leftarrow v + 1$
  $rcvd.c_v \leftarrow c$
  $(\mathsf{ad}, m, \alpha, st_{\mathrm{D}}) \leftarrow \mathrm{D}(k, \mathsf{ad}, c, st_{\mathrm{D}})$
  if $(\alpha = 1) \wedge \mathsf{cond}_i$ then
      $\mathsf{phase} \leftarrow 1$
  end if
  if $\mathsf{phase} = 1$ then
      return $m$
  end if
  return $\perp$
```

1. **Basic authenticated encryption:**
   $\mathsf{cond}_1 = (\nexists w : (c = sent.c_w) \wedge (\mathsf{ad} = sent.ad_w))$
2. **Basic authenticated encryption, no replays:**
   $\mathsf{cond}_2 = (\nexists w : (c = sent.c_w) \wedge (\mathsf{ad} = sent.ad_w)) \vee (\exists w < v : c = rcvd.c_w)$
3. **Basic authenticated encryption, no replays, strictly increasing:**
   $\mathsf{cond}_3 = (\nexists w : (c = sent.c_w) \wedge (\mathsf{ad} = sent.ad_w)) \vee (\exists w, x, y : (w < v) \wedge (sent.c_x = rcvd.c_w) \wedge (sent.c_y = rcvd.c_v) \wedge (x \geq y))$
4. **Basic authenticated encryption, no replays, strictly increasing, no drops:**
   $\mathsf{cond}_4 = (u < v) \vee (c \neq sent.c_v) \vee (\mathsf{ad} \neq sent.ad_v)$

SECURE CHANNELS WITH TLS

- Paterson–Ristenpart–Shrimpton 2011

$$\left.\begin{array}{l}\text{MEE–TLS encoding – CBC}\\ \text{(message len.)} + \text{(tag len.)} > \text{(block len.)} - 8\end{array}\right\}\ \text{TLS satisfies }\textbf{Level 1}\text{ AEAD}$$

SECURE CHANNELS WITH TLS

Authenticated and Confidential Channel Establishment (ACCE)

- Jager–Kohlar–Schäge–Schwenk 2012

Stateful length-hiding AEAD at **Level 4** }

ACCE security for TLS
(Suites: TLS-DHE)

- Krawczyk–Paterson–Wee 2013

Stateful length-hiding AEAD at **Level 4**
Constrained chosen ciphertext security }

ACCE security for TLS
(Suites: TLS-RSA,
TLS-CCA, TLS-DH, TLS-DHE)

IMPLICATIONS BETWEEN AUTHENTICATION LEVELS

st'_{E} and st'_{D}:

- st'_{E} : st'_{E}.substate := st_{E}, where st_{E} is the state in Π, st'_{E}.counter
- st'_{D} : st'_{D}.substate := st_{D}, where st_{D} is the state in Π, st'_{D}.status, st'_{D}.sqnlist

$\mathrm{Kgn}'()$:

```
return  $\Pi.\mathrm{Kgn}()$
```

$\mathrm{Snd}'(k, m, st'_{\mathrm{E}})$:

```
$(c, st'_{\mathrm{E}}.\text{substate})$
    $\leftarrow \Pi.\mathrm{Snd}(k, \boxed{\texttt{Ecd}(st'_{\mathrm{E}}.\text{counter}, m)}, st'_{\mathrm{E}}.\text{substate})$
$st'_{\mathrm{E}}.\text{counter} \leftarrow st'_{\mathrm{E}}.\text{counter} + 1$
return  $(c, st'_{\mathrm{E}})$
```

$\mathrm{Rcv}'(k, c, st'_{\mathrm{D}})$:

```
if $st'_{\mathrm{D}}.\text{status} = \text{failed}$ then
    return  $(\perp, 0, st_{\mathrm{D}})$
end if
$(m_\Pi, \alpha, st'_{\mathrm{D}}.\text{substate})$
    $\leftarrow \Pi.\mathrm{Rcv}(k, c, st'_{\mathrm{D}}.\text{substate})$
if $\alpha = 1$ then
    $\boxed{(\text{sqn}, m, \alpha) \leftarrow \texttt{Dcd}(st'_{\mathrm{D}}.\text{sqnlist}, m_\Pi)}$
end if
if $(\alpha = 0) \vee \boxed{\texttt{TEST}i}$ then
    $st'_{\mathrm{D}}.\text{status} = \text{failed}$
    return  $(\perp, 0, st'_{\mathrm{D}})$
end if
$st'_{\mathrm{D}}.\text{sqnlist} = st'_{\mathrm{D}}.\text{sqnlist} || \text{sqn}$
return  $(m, \alpha, st'_{\mathrm{D}})$
```

- **Basic authentication, no replays:**
  $\texttt{TEST2} = (\exists j : \text{sqn} = st'_{\mathrm{D}}.\text{sqnlist}_j)$
- **Basic authentication, no replays, strictly increasing:**
  $\texttt{TEST3} = (\exists j : \text{sqn} \not> st'_{\mathrm{D}}.\text{sqnlist}_j)$
- **Basic authentication, no replays, strictly increasing, no drops:**
  $\texttt{TEST4} = (\exists j : \text{sqn} \not> st'_{\mathrm{D}}.\text{sqnlist}_j) \vee (\text{sqn} \neq \max\{st'_{\mathrm{D}}.\text{sqnlist}_j\} + 1)$

ANALYSIS METHOD

Computational Analysis:

Complexity-theoretic reduction proofs

- Protocol specification
- Adversary capabilities
- Adversary winning conditions

Security is reducible to that of an underlying *hard* problem

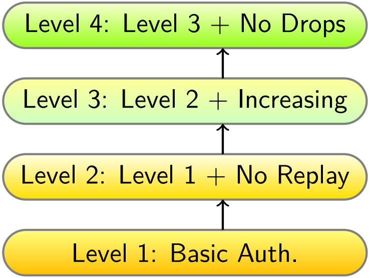
IMPLICATIONS BETWEEN AUTHENTICATION LEVELS

Theorem

Let Π be a secure level-1 authentication scheme and $\texttt{Coding}$ be an authentication encoding scheme with collision-resistant encoding. Let $i \in \{2, 3, 4\}$. Then $\Pi'_i = P(\Pi, \texttt{Coding}, \texttt{TEST}i)$ is a secure level-i authentication scheme. Specifically, let $\mathcal{A}$ be an adversary algorithm that runs in time t and asks q_s Send *queries and q_r* Recv *queries, and let $q = q_s + q_r$. Then there exists an adversary $\mathcal{B}$ that runs in time $t_\mathcal{B} \approx t$ and asks no more than $q_\mathcal{B} = \frac{1}{2} q_s(q_s - 1)$ queries, and an adversary $\mathcal{F}$ that runs in time $t_\mathcal{F} \approx t$ and asks $q_\mathcal{F} = q$ queries, such that*

$$\mathbf{Adv}^{\mathsf{auth}_i}_{P(\Pi, \texttt{Coding}, \texttt{TEST}i)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{auth}_1}_{\Pi}(\mathcal{F}) + \mathbf{Adv}^{\mathsf{collision}}_{\texttt{Ecd}}(\mathcal{B}) \ .$$

Implications between Authentication Levels



Level 4: Level 3 + No Drops — Received sqn is exactly the next expected one

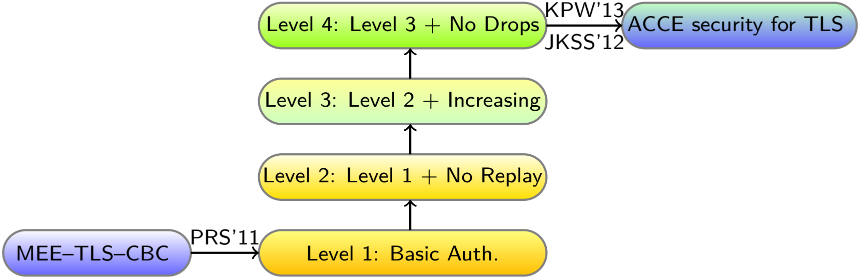
Level 3: Level 2 + Increasing — Received sqn is strictly greater than the last received

Level 2: Level 1 + No Replay — Sequence number sqn does not repeat

Level 1: Basic Auth.

Sequence number can be included **implicitly** or **explicitly**

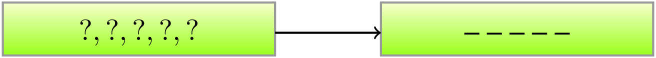
AUTHENTICATION LEVELS APPLIED – TLS


Level 4: Level 3 + No Drops
KPW'13
JKSS'12
ACCE security for TLS
Level 3: Level 2 + Increasing
Level 2: Level 1 + No Replay
MEE–TLS–CBC
PRS'11
Level 1: Basic Auth.

AUTHENTICATION LEVELS APPLIED

- Protocol Analysis
  - Selection of appropriate authentication experiment
- Building Authentication Protocols
  - Encoding and checking sequence numbers to achieve desired level

Questions



?, ?, ?, ?, ?

– – – – –