IDENTIFYING POTENTIAL TYPE CONFUSION ATTACKS IN AUTHENTICATED MESSAGES

Catherine Meadows
Code 5543
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375
meadows@itd.nrl.navy.mil
http://chacs.nrl.navy.mil

WHAT IS A TYPE CONFUSION ATTACK?

- Attack on a protocol that relies on a principal's inability to distinguish between two strings of two different types
  - Hard to verify that type confusion attacks are realistic
  - Depends on formatting of data
- What we show in this talk
  - Example of a realistic type confusion attacks
  - Formal model of type confusion
  - Outline of a technique for detecting type confusion

A SIMPLE TYPE CONFUSION ATTACK

Needham–Schroeder public key protocol

1. A –> S: B
2. S –> A: $\{K_B,B\}Sig_S$
3. A –> B: $\{R_A,A\}K_B$
4. B –> S: A
5. S –> B: $\{K_A,A\}Sig_S$
6. B –> A: $\{R_A,R_B\}K_A$
7. A –> B: $\{R_B\}K_B$

THE “ATTACK”

3. I_A –> B: $\{R_I,A\}K_B$

4. B –> S: A

5. S –> B: $\{K_A,A\}Sig_S$

6. B –> A: $\{R_I,R_B\}K_A$

   I intercepts this message

3’. I_{RB} –> A: $\{R_I,R_B\}K_A$

   I sends the intercepted message to A as an initiator’s message, with R_B as the name field

4’. A –> S: R_B

   A sends the “name” R_B to S in order to get its public key

7. I_A –> B: $\{R_B\}K_B$

   I now has the information it needs to impersonate A to B. It encrypts R_B with K_B and sends it to B

HOW TO PREVENT IT

- **These type of attacks are easy to prevent, in principle**
- **Any implementation of a protocol will generally include fields that describe the type of data that is being passed in them**
- **Heather, Lowe, and Schneider have shown that, assuming the Dolev–Yao attacker model, labelling is enough to prevent type confusion attacks**
  - **Dolev-Yao model**
    - Intruder who can read, alter, intercept traffic, perform crypto operations
    - May be in league with dishonest principals
    - Can not break cryptosystem or guess keys
- **Are there any realistic examples of type confusion attacks?**

A MORE REALISTIC TYPE CONFUSION ATTACK

- Found in Group Domain of Interpretation (GDOI) protocol
- Protocol facilitating distribution of group keys by Group Controller Key Server (GCKS)
- Based on ISAKMP and IKE
  - Standards developed for key exchange between two principals
- GDOI uses
  - IKE to distribute pairwise keys
  - Groupkey Pull Protocol initiated by member to distribute Key Encryption Keys (KEKs) to new group member
  - Groupkey Push Message to distribute KEK and Traffic Encryption Keys (TEKs) to existing group members
- GDOI uses ISAKMP formatting conventions
  - Using conventions designed for pairwise protocols in group protocols caused some problems, as it turned out

GDOI PROTOCOLS

Groupkey Pull Protocol

```
     Initiator (Member)           Responder (GCKS)
     ------------------           ----------------
     HDR*, HASH(1), Ni, ID    -->
                       <--    HDR*, HASH(2), Nr, SA
     HDR*, HASH(3) [, KE_I]   -->
         [,CERT] [,POP_I]
                       <--    HDR*, HASH(4), [KE_R,] SEQ,
                                       KD [,CERT] [,POP_R]
```

CERT used to verify information about principals
POP used to verify possession of key provided in CERT
POP = signature on hash of Ni,Nr
HDR begins with random numbers contributed by both principals

Groupkey Push Message

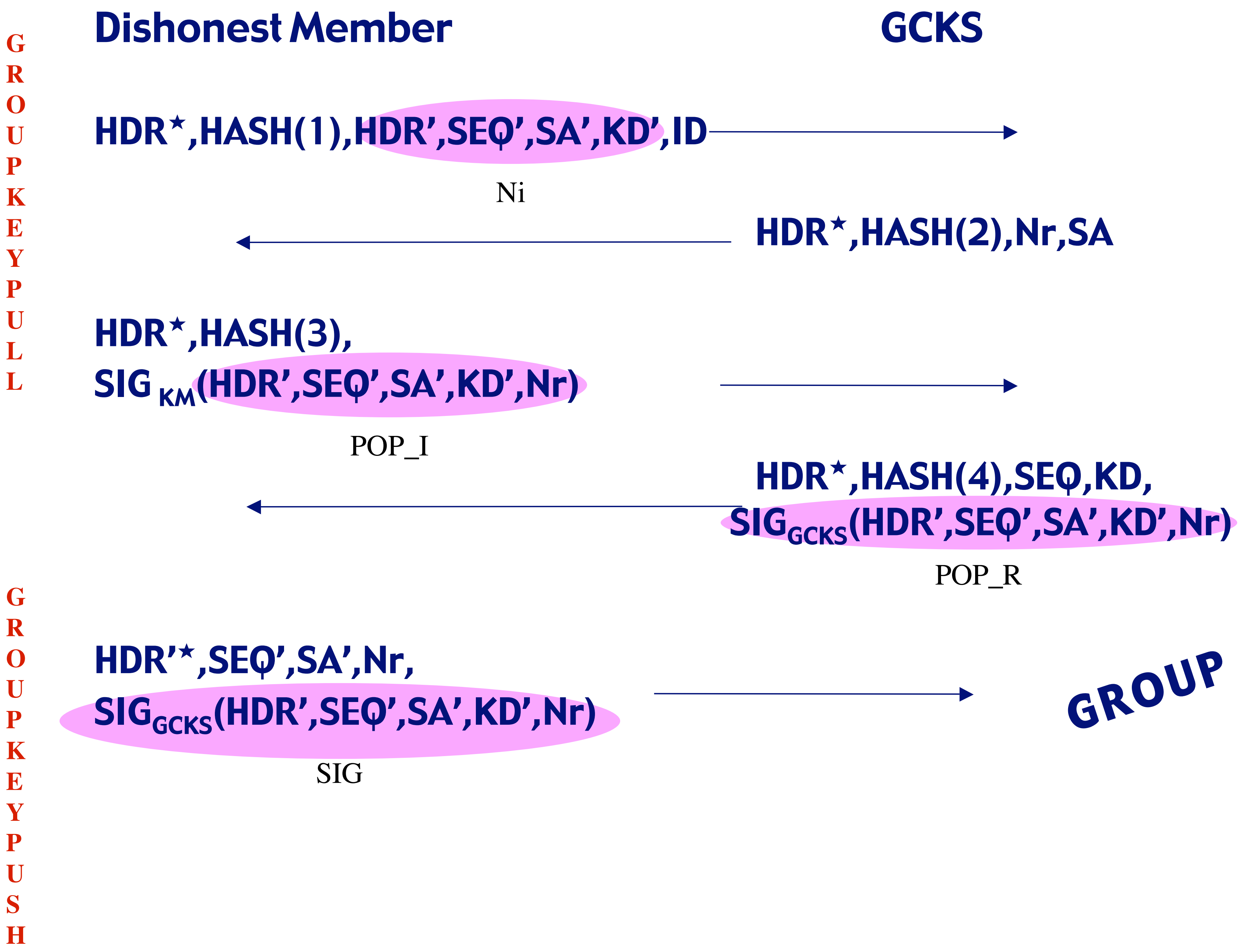
```
         Member                   GCKS or Delegate
     ------                       ----------------

                   <-----HDR*, SEQ, SA, KD, [CERT,] SIG
```

HDR begins with random number supplied by GCKS
All other message fields identified by type and length labels
Signature taken over hash of entire message (before encryption), including header
KD is likely to end in a random number

POSSIBLE ATTACK

- In third message of groupkey pull protocol, GCKS signs (N_A, N_B)
  - N_A member's nonce, N_B the GCKS's nonce
- Suppose
  - Group member dishonest
- Then …



GROUPKEYPULL

Dishonest Member

GCKS

HDR*,HASH(1),HDR’,SEQ’,SA’,KD’,ID →

Ni

← HDR*,HASH(2),Nr,SA

HDR*,HASH(3),
SIG_{KM}(HDR’,SEQ’,SA’,KD’,Nr) →

POP_I

← HDR*,HASH(4),SEQ,KD,
SIG_{GCKS}(HDR’,SEQ’,SA’,KD’,Nr)

POP_R

GROUPKEYPUSH

HDR’*,SEQ’,SA’,Nr,
SIG_{GCKS}(HDR’,SEQ’,SA’,KD’,Nr) →

SIG

GROUP

HOW TO AVOID THESE

- **Require principals not to accept signed message unless contains nonce contributed by that principal**
  - **Not an option for group protocols**
- **Require different mechanisms for different apps.**
  - **Examples**
    - Different signature keys for POP and groupkey
    - Different hash function
    - Keyed hash function with different keys
  - **Often the best option, but too much reliance on this could lead to intractable key management problem**
- **Solution chosen: take signature over message and label indicating type of message**
  - **Similar to solution recommended by Heather, Lowe and Schneider**

WHY WORRY ABOUT TYPE CONFUSION WHEN SEEMS EASY TO AVOID IT?

- Can design protocols so that they are free of type confusion, but what about protocols they interact with?
- A similar type confusion attack we found with NPA relied upon interaction with ISAKMP, protocol GDOI built on top of
- Type confusion analysis will help us determine whether or not it is safe for two protocols to interact
  - Carrying Herzog and Guttman’s work one step further
  - They show protecting against harmful interaction boils down to protecting against confusion between messages

A CLOSER LOOK: WHEN IS THE GDOI ATTACK POSSIBLE?

- Depends upon relationship between types and lengths of data used in the two messages
- First message contains two random strings Ni and Nr
  - Only Nr recognizable by message generator
- Second message contains a string of recognizable and random fields F, followed by a random string S
- Constraints are
  - Length(Ni) ≥ Length (F)
  - Length(Nr) ≤ Length(S)
  - Length(Ni) + Length(Nr) = Length(F) + Length(S)

SOLUTION WILL HAVE THREE PARTS

- Definition of
  - type
  - type local to a principal
  - type under the control of a principal
- Definition of game between intruder and honest principals
- Procedure for determining whether or not intruder can win, and what is the probability of winning

DEFINITION OF TYPE

- A type is either a set of bit-strings, or a probability function whose domain is a set of bit-strings
- A type member choice is the act of choosing a member of that type
- We say that a type is under control of a principal A if A is the principal who performs the type member choice
- We say that a type is local to A if A is able to verify membership in the type

WHAT MESSAGES LOOK LIKE

- Masquerading message constructed by an honest principal A
  - Possibly containing information supplied by the intruder
- Spoofed message expected by an honest principal B
- From A’s point of view, masquerading message constructed from
  - Types controlled by A
    - Data it constructed itself
  - Types controlled by other honest principals
    - Data constructed by other honest principals whose origin it can verify
  - Types directly controlled by the intruder
    - Data generated by the intruder
- From B’s point of view spoofed message constructed from
  - Types controlled by B
    - Data it constructed itself that it is expecting to see in the message
  - Types controlled by other honest principals
    - Data constructed by others received previously, that it is expecting to see in the message
  - Types directly controlled by the intruder
    - Data constructed by the intruder that B received previously
  - Types indirectly controlled by the intruder
    - Data B is seeing for the first time

EXAMPLE

1. A –> B : N_A

   N_A nonce of length N
2. B –> A: N_B, $S_B(N_A, N_B)$

   N_B nonce of length N
3. A –> B: $S_A(N_B,N'_A)$

   N'_A nonce of length N

1’. B –>A : N''_B

2’. A –> B: N''_A, $S_A(N''_B, N''_A)$

3’. B –> A: $S_B(N''_A, N'''_B)$

- **Let ‹ X, N_B › from line 2 be a masquerading message sent from B to A in the first instance of the protocol**
  - N_B
    - Set of bit-strings of length N with uniform distribution
    - Is under control of B
  - X
    - Set of bit-strings of length N
    - Is under the direct control of the intruder
- **Let ‹ $N''_{A,}$ Y › from line 3 be a spoofed message from B to A second instance**
  - N''_A
    - Set of bit-strings of length N with uniform distribution
    - Is under control of A
  - Y
    - Set of bit-strings of length N
    - Is under the indirect control of the intruder

DEFINITION OF TYPE FUNCTION TREE

- Expressed as a function because choice of later fields may be influenced by choice of earlier fields
- A type function tree is a function $\mathscr{R}$ from lists of bit-strings to types, such that:
  1. The empty list <> is in the domain of $\mathscr{R}$
  2. A list of bit-strings $<x_1,\ldots,x_k>$ is in the domain of $\mathscr{R}$ if and only if
     - $< x_1,\ldots,x_{k-1}>$ □dom($\mathscr{R}$)
     - x_k □ $\mathscr{R}(< x_1,\ldots,x_{k-1}>)$)
  3. There exists an integer h, called the height of $\mathscr{R}$, such that for any n > h, $\mathscr{R}(< x_1,\ldots,x_n>) = \{$□$\}$, where □is the empty string
- We let $\mathscr{R}^k$ denote the restriction of $\mathscr{R}$ to k-tuples
- Type function tree reflects temporal and causal order of choice of fields in a message

EXAMPLE (AGAIN)

1. A –> B : N_A
   N_A nonce of length N
2. B –> A: $N_B, S_B(N_A, N_B)$
   N_B nonce of length N
3. A –> B: $S_A(N_B, N'_A)$

1'. B –>A : N''_B

2'. A –> B: $N''_A, S_A(N''_B, N''_A)$

3'. B –> A: $S_B(N''_A, N'''_B)$

- Let ‹ X, N_B › from line 2 be the masquerading message
- Let ‹ $N''_{A,}$ Y › from line 3 be the spoofed message
- Two type function trees

  Corresponding to < X, N_B >
  $\mathscr{R}(<>) = X$ 2. $\mathscr{R}(<x_1>) = N_B$
  Corresponding to < N'_A ,Y>
  1. $\mathscr{S}(<>) = N''_A$ 2. $\mathscr{S}(<x_1>) = Y$

INTERLEAVING TYPE FUNCTION TREES

- Let $\mathscr{R}_1$ and $\mathscr{R}_2$ be two type function trees of height h_1 and h_2, respectively
- Define an interleaving $\mathscr{I}$ of $\mathscr{R}_1$ and $\mathscr{R}_2$ as follows
  - Let $\square_1$ and $\square_2$ be monotone increasing injections of $<1,\ldots,h_1>$ and $<1,\ldots,h_2>$ into $<1,\ldots,h>$ such that each member of $<1,\ldots,h>$ is in the image of $\square_1$ or $\square_2$
  - If $1 = \square_i(t)$ for some t, we let $\mathscr{I}(<>) = \mathscr{R}i(<>)$
  - If $<x_1,\ldots,x_k>$ is in the domain of $\mathscr{I}$, and k+1 is in the image of $\square_i$, we let $\mathscr{I}(<x_1,\ldots,x_k>) = \mathscr{R}i\ (<x_{j1},\ldots,x_{jt}>)$ where $<j_1,\ldots,j_t>$ is the maximal subsequence of $<1,\ldots, k>$ in the image of $\square_i$
- An interleaving of two different type function trees is also a type function tree if $\square_1$ and $\square_2$ are disjoint or if the two type function trees agree on some initial common data

EXAMPLE (AGAIN)

1. A –> B : N_A

   N_A nonce of length N

2. B –> A: N_B, $S_B(N_A, N_B)$

   N_B nonce of length N

3. A –> B: $S_A(N_B,N'_A)$

1'. B –>A : N''_B

2'. A –> B: N''_A, $S_A(N''_B, N''_A)$

3'. B –> A: $S_B(N''_A, N'''_B)$

‹ X, N_B ›(line 2) masquerading message & ‹ $N''_{A,}$Y › (line 3) spoofed message

Two type function trees

Corresponding to < X, N_B >

1. $\mathscr{R}$(<>) = X. 2. $\mathscr{R}$(<x_1>) = N_B

Corresponding to < N'_A ,Y>

1. $\mathscr{S}$(<>) = N''_A 2.$\mathscr{S}$(<x_1>) = Y

Some possible interleavings

Case 1: N''_A learned by intruder before X and N_B before Y

1. $\mathscr{A}$(<>) = N''_A. 2. $\mathscr{A}$(<x_1>) = X 3. $\mathscr{A}$(<x_1,x_2>) = N_B 4. $\mathscr{I}$(<x_1,x_2,x_3>) = Y

Case 2: N''_A learned by intruder after X and N_B before Y

1. $\mathscr{A}$(<>) = X 2. $\mathscr{I}$(<x_1>) = N_B 3. $\mathscr{I}$(<x_1,x_2>) = N''_A 4. $\mathscr{I}$(< x_1,x_2,x_3 >) = Y

MESSAGE TYPE

- Let $\mathscr{R}$ be a type function tree of height h
- Let □ be a map from some ‹1,…,q› onto ‹1,…,h›
- We say that M is a **message type constructed from $\mathscr{R}$ via □** if
  - M consists of all fields of the form y_1 || … || y_q such that there exists an <x_1,…,x_h> in the domain of $\mathscr{R}$ h such that $y_j = x_i$ whenever j = □(i)
- We call □ a **message surjection**
- Message surjections allow us to capture the fact that order of appearance of data in a message does not reflect order in which it was generated

A GAME BETWEEN INTRUDER AND HONEST PRINCIPALS

- Fix on an order for choosing members of types by honest principals and intruder
- Have honest principals choose members according to the rules of the protocol
- Have intruder choose members in according to a strategy that maximizes the probability of a successful masquerade
- Winning strategy is one that puts probability of success above a certain threshold

EXAMPLE (again)

1. A –> B : N_A

   N_A nonce of length N

2. B –> A: N_B, $S_B(N_A, N_B)$

   N_B nonce of length N

3. A –> B: $S_A(N_B, N'_A)$

1'. B –>A: N''_B

2'. A –> B: N''_A, $S_A(N''_B, N''_A)$

3'. B –> A: $S_B(N''_A, N'''_B)$

- Let ‹ X, N_B › from line 2 be the masquerading message
- Let ‹ $N''_{A,}$ Y › from line 3 be the spoofed message

Case 1: N'_A learned by intruder before X or and N_B before Y

1. $\mathcal{I}(<>) = N''_A$. 2. $\mathcal{A}(<x_1>) = X$ 3. $\mathcal{A}(<x_1,x_2>) = N_B$ 4. $\mathcal{A}(<x_1,x_2,x_3>) = Y$

A chooses x1 randomly from N''_A, B chooses x3 randomly from N_B,

Winning strategy: choose x2 = N''_A and x4 = N_B

Case 2: Suppose N''_A learned by intruder after X and N_B before Y

1. $\mathcal{I}(<>) = X$ 2. $\mathcal{I}(<x_1>) = N_B$ 3. $\mathcal{I}(<x_1,x_2>) = N''_A$ 4. $\mathcal{A}(< x_1,x_2,x_3 >) = Y$

Intruder chooses x1 from X, A chooses x2 randomly from N''_A, B chooses x3 randomly from N_B

Best strategy: choose any member of X, and choose $x_4 = x_3$

Probability of spoofing = $1/2^N$

HOW TO COMPUTE PROBABILITY OF SUCCESS IN GENERAL CASE

- Use the fact that intruder's success in inducing type confusion will depend on which types he tries to match with each other
- This in turn introduces constraints on lengths of fields in respective messages
- What we need: a list of the possible constraints

CONSTRAINT TREES

Let $\langle i_1, \ldots, i_m \rangle$ and $\langle j_1, \ldots, j_n \rangle$ be two sets of indices

We construct a constraint tree as follows

1. Root of constraint tree is the empty set
2. Children of the root are

   $C1 = \{l(x_{i(1))} \leq l(x_{j(1)})\}$

   $C2 = \{l(x_{i(1)}) > l(x_{j(1)}),\ l(x_{i(1)}) \leq l(x_{j(1)}) + l(x_{j(2)})\}$,

   …,

   $Cn\text{-}1 = \{l(x_{i(1)}) > l(x_{j(1)}) + \ldots + l(x_{j(n-1)}),\ l(x_{i(1)}) \leq l(x_{j(1)})) + \ldots + l(x_{j(n)})\}$
3. If D at s'th level is a node containing $l(x_{i(1)}) + \ldots + l(x_{i(s)}) < l(x_{j(1)}) + \ldots + l(x_{j(t)})$, construct children as follows

   $D1 = D \cup \{l(x_{i(1)}) + \ldots + l(x_{i(s+1)}) \leq l(x_{j(1)}) + \ldots + l(x_{j(t)})\}$,

   $D2 = D \cup \{l(x_{i(1)}) + \ldots + l(x_{i(s+1)}) > l(x_{j(1)}) + \ldots + l(x_{j(t)}),\ l(x_{i(1)}) + \ldots + l(x_{i(s+1)}) \leq l(x_{j(1)}) + \ldots + l(x_{j(t+1)})\}$,

   …,

   $Dn\text{-}1 = D \cup \{l(x_{i(1)}) + \ldots + l(x_{i(s+1)}) > l(x_{j(1)}) + \ldots + l(x_{j(n-1)}),\ l(x_{i(1)}) + \ldots + l(x_{i(s+1)}) \leq l(x_{j(1)}) + \ldots + l(x_{j(n)})\}$

THE GAP–TOOTHED ZIPPER

Let $\mathscr{R}$ and $\mathscr{S}$ be two type function trees of height h_1 and h_2

Let $\square_1$ be a function from $\langle 1,\ldots,t_1 \rangle$ onto $\langle 1,\ldots,h_1 \rangle$ constructing a masquerading message

Let $\square_2$ be a function from $\langle 1,\ldots,t_2 \rangle$ onto $\langle 1,\ldots,h_2 \rangle$ constructing a spoofed message

Let $\mathscr{I}$ be an interleaving of $\mathscr{R}$ and $\mathscr{S}$ constructed using injections $\square_1$ and $\square_2$

Let p be a number between 0 and 1

We construct a gap–toothed zipper $\boldsymbol{Z}$ ($\mathscr{I}$,p) as follows

HOW TO CONSTRUCT THE GAP–TOOTHED ZIPPER

- **To each leaf node C of the constraint tree, add the equation E =**

  $x_{\square 1 \square 1}(1) \,||\, \ldots \,||\, x_{\square 1 \square 1}(t_1) = x_{\square 2 \square 2}(1) \,||\, \ldots \,||\, x_{\square 2 \square 2}(t_2)$
- **For each set of constraints B constructed as above**
  - **Choose members of types consistent with B in the order they appear in $\mathscr{I}$**
  - **Choose according to defined probability distribution if under the control of honest principals**
  - **Choose according to some distribution d if under the direct control of the intruder**
  - **After each choice, check whether probability E satisfied given C satisfied is less than p**
    - If it is, discard the sequence of choices
    - If not, continue choosing members of types

EXAMPLE

1. A –> B : N_A

   N_A nonce of length N

2. B –> A: N_B, $S_B(N_A, N_B)$

   N_B nonce of length M

3. A –> B: $S_A(N_B, N'_A)$

   N'_A nonce of length N'

1'. B –>A : N''_B

N''_b nonce of length N

2'. A –> B: N''_A, $S_A(N''_B, N''_A)$

N''_A nonce of length M

3'. B –> A: $S_B(N''_A, N'''_B)$

N'''_B nonce of length N'

Masquerading message: 2nd message (X, N_B)

X of length N, N_B of length M

Spoofed message: 3rd message (N''_A, Y)

N''_A of length M, Y of length N' (= N by length constraints)

Order of choice: $x_1 = N''_A$, $x_2 = X$, $x_3 = N_B$, $x_4 = Y$

Equality constraint $x_2 \,||\, x_3 = x_1 \,||\, x_4$

Look at the following length constraint

$C_1 = \{l(x_2) < l(x_1), l(x_2) + l(x_3) = l(x_1) + l(x_4)\}$

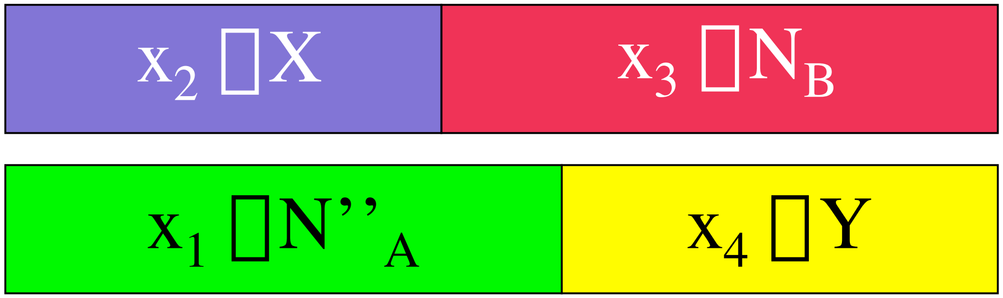
Assume N and M chosen so that C sat., e.g. N < M

The Zipper in Action

Order of choice: $x_1 \square N''_A$, $x_2 \square X$, $x_3 \square N_B$, $x_4 \square Y$

Equality constraint: $x_2 \,||\, x_3 = x_1 \,||\, x_4$

Length constraint: $\{l(x_2) < l(x_1),\ l(x_2) + l(x_3) = l(x_1) + l(x_4)\}$



Choose x_1. Probability x_1 consistent with constraints is 1

Choose x_2. Probability $\langle x_1,x_2 \rangle$ consistent with constraints is 1 if choose x_2 equal to first $l(x_2)$ bits of x_1, 0 otherwise

Choose x_3. Prob. $\langle x_1,x_2,x_3 \rangle$ consistent with constraints is $1/2^{(l(x1)-l(x2))}$

Choose x_4. Cond. Prob. of sat constraints is 1 if choose x_4 equal to last $l(x_4)$ bits of x_3, 0 otherwise

Final probability: $1/2^{(l(x1)-l(x2))}$

CONCLUSIONS AND DISCUSSION

- **Have developed methodology for identifying type confusion**
  - **Takes into account possibility of confusing pieces of fields with each other as well as entire fields**
  - **Takes into account probabilistic nature of type confusion**
- **Some possible extensions**
  - **Type function trees of unbounded height**
    - Especially, unbounded repetitions of the same type
  - **Applying technique to encryption functions as well**
    - Type confusion attacks found by Bellovin on ESP
      - ESP did not distinguish between beginning, middle, and end of a message
      - Allowed for various types of truncation and cut-and-paste attacks
    - Extra difficulty in encryption: encrypted message will have type of message if one knows the key, type of random nonces if not