COL876: SPECIAL TOPICS IN FORMAL METHODS

Formal verification of security protocols

Lecture 5, 10 August 2023

APPLIED-PI CALCULUS: GRAMMAR

P, Q :=	plain process
0	[null process]
P | Q	[parallel composition]
!P	[replication]
νn.P	[name restriction]
if $t_1 = t_2$ then P else Q	[conditional branching]
in(c, x).P	[receive action]
out(c, t).P	[send action]
let x = t in P	[let binding]

FORMALIZING EXECUTIONS

$$A \longrightarrow B : A, \mathsf{enc}(m, \mathsf{pk}(B))$$
$$B \longrightarrow A : \mathsf{enc}(m, \mathsf{pk}(A))$$

```
init(ski: skey, pkr: pkey) {
    new n: bytes;
    send(pk(ski), aenc(n,pkr));
    recv(x: bytes);
    if (adec(x,ski) ≠ n)
      error;
}
```

```
resp(skr: skey) {
    recv(k: pkey, y: bytes);
    let
        z = adec(y, skr)
    in
        send(aenc(z,k));
}
```

APPLIED-PI FORMALISM

- $P_i(ski, pkr) \triangleq \nu n.\ \mathrm{out}(c, \mathrm{aenc}(n, pkr)).\ \mathrm{in}(c, x).\ \mathrm{if}(\mathrm{adec}(x, ski) == n)$ then SUCCESS
- $P_r(skr) \triangleq \mathrm{in}(c, y).$ let $pka = \mathrm{fst}(y)$ in. let $z = \mathrm{adec}(y, skr)$ in. $\mathrm{out}(c, \mathrm{aenc}(z, pka))$
- Allow the intruder to supply the other party the initiator talks to
- Allow the same agent to play either role; allow unboundedly many honest agents
- Can write this out more succinctly as follows:

$$\mathrm{Pr} \triangleq\ !\nu sk.(\ !\mathrm{in}(c, x_{pk}).\ P_i(sk, x_{pk})\ |\ !P_r(sk)\ |\ \mathrm{out}(c, \mathrm{pk}(sk))\ )$$

SECRECY

- n should be secret to the initiator and the responder
- Is there *any* session where the name established between the initiator and responder in *that* session can be deduced by the intruder?
- $Q_i(n, ski, pkr) \triangleq \text{out}(c, \text{aenc}(n, pkr)).\ \text{in}(c, x).\ \text{if}(\text{adec}(x, ski) == n) \text{ then SUCCESS}$, and $P_i(ski, pkr) \triangleq \nu n.\ Q_i(n, ski, pkr)$
- $P^n(ski, pkr) = \nu n.\ (Q_i(n, ski, pkr) \mid (\text{in}(c, x).\ \text{if } x = n \text{ then event } leak(n) \text{ else } 0)$
- $Pr^s \triangleq \text{in}(c, x_{pk}).\ P^n(s_i, x_{pk}) \mid !\nu sk.\ (!\text{in}(c, x_{pk}).\ P_i(sk, x_{pk}) \mid !P_r(sk) \mid \text{out}(c, pk(sk)))$

EXAMPLE 2: NS PUBLIC KEY

$$A \longrightarrow B : \mathsf{enc}((A, n_a), \mathsf{pk}(B))$$

$$B \longrightarrow A : \mathsf{enc}((n_a, n_b), \mathsf{pk}(A))$$

$$A \longrightarrow B : \mathsf{enc}(n_b, \mathsf{pk}(B))$$

EXAMPLE 2: NS PUBLIC KEY

$$A \rightarrow B : \mathsf{enc}((A, n_a), \mathsf{pk}(B))$$
$$B \rightarrow A : \mathsf{enc}((n_a, n_b), \mathsf{pk}(A))$$
$$A \rightarrow B : \mathsf{enc}(n_b, \mathsf{pk}(B))$$

```
init(ski: skey, pkr: pkey) {
    new na: bytes;
    send(aenc((pk(ski), na), pkr));
    recv(x: bytes);
    let z = adec(x, ski) in
      if (fst(z) =/= na) error
      else send(aenc(snd(z), pkr);
}
```

```
resp(pki: pkey, skr: skey) {
    recv(y: bytes);
    let (k, na) = adec(y, skr) in
        new nb: bytes;
        send(aenc((na, nb), k));
        recv(z: bytes);
        if (adec(z, skr) =/= nb) error;
}
```

NS PUBLIC KEY

- Proposed by Roger Needham and Michael Schroeder in 1978.
- Requirement: At the end of an execution, the two agents should agree on the identity of their respective correspondent.
- Is there an attack?
-

NS PUBLIC KEY

- Proposed by Roger Needham and Michael Schroeder in 1978.
- Requirement: At the end of an execution, the two agents should agree on the identity of their respective correspondent.
- Is there an attack?
- Yes! Found by Gavin Lowe in 1995. Different flavour of MitM.

CORRESPONDENCE: FORMALIZED

- $e_0(\overrightarrow{t_0}) \triangleright e_1(\overrightarrow{t_1})$ denotes the following correspondence: "if $e_1(\overrightarrow{t_1})$ occurred in a run, then $e_0(\overrightarrow{t_0})$ occurred earlier"
- A reduction sequence $P_0 \xrightarrow{\gamma_1} P_1 \cdots \xrightarrow{\gamma_n} P_n$ satisfies a correspondence $e_0(\overrightarrow{t_0}) \triangleright e_1(\overrightarrow{t_1})$ iff for any σ,

  whenever $e_1(\overrightarrow{t_1}\sigma)$ occurs in some P_i, there is a $j \leq i$ such that $e_0(\overrightarrow{t_0}\sigma)$ occurs in P_j
- A process P satisfies a correspondence property iff all reduction sequences starting from P satisfy it.

TOY VOTING PROTOCOL

- Consider a really simple voting protocol
- Voter V encrypts their vote v in C's public key and sends it
- C decrypts it and counts the vote
- *Anonymity*: nobody but C should be able to find a link between V's name and their vote
- Is there an attack? What does it mean to find a link?

ANONYMITY

- Consider a situation where only V has voted, nobody else
- The intruder sees a single term going by on the channel
- The term is aenc(v, pk(A))
- Can the intruder make any judgements based on this one term?

ANONYMITY

- Assume the set of candidates is $\{v_0, v_1\}$ (both public constants)
- What would differ between a situation where V voted for v_0 versus one where they voted for v_1?
-

ANONYMITY

- Assume the set of candidates is $\{v_0, v_1\}$ (both public constants)
- What would differ between a situation where V voted for v_0 versus one where they voted for v_1?
- Frames $\sigma_0 = [x \mapsto aenc(v_0, pk(A))]$ and $\sigma_1 = [x \mapsto aenc(v_1, pk(A))]$
- What recipe tells these frames apart?