Ubiquitous Computing: Security

Topic 6

Stajano and Ross's Resurrecting Duckling Protocol

Consider a Ubicomp Environment with heterogeneous nodes

- Some nodes have more power (e.g. cell phones/lap top)
- Others have very limited embedded computing

Stajano and Ross assume a wireless ad hoc network

Security goals

- Availability - is the node on hand when needed
- Authenticity - to whom can a principle talk?
- Integrity - ensuring that a node is not maliciously altered
- Confidentiality - employ appropriate discretion with secrets

The constraints on the embedded nodes are:

- Peanut CPU (very limited processing)
- Battery Power
- High Latency

Hence the embedded nodes may not be able to use strong cryptography.

Availability

Availability is perhaps the most important constraint

- Since if the device is not available, do the other issues matter?
- Classical attack of radio frequency jamming
- New attack - sleep deprivation torture attack
  - Listening for radio signals has high power consumption
  - Nodes need to sleep most of the time to conserve battery life
  - So attacker could send malicious frames in an attempt to get the nodes for (nearly) continuous listening
  - Defend by prioritizing tasks and reserving power (much like QoS)

Authenticity

Traditional approaches may use trusted third party

- e.g. key escrow
- Such approaches are not appropriate in ad hoc networks

So how can devices recognize the owner/administrator?

- Assume that owner has a sort of "Universal Remote Control" which he keeps secured
- Secure transient association used when deploying devices
- Imprinting used to establish shared secret
  - Lorenz's ducklings which treat the first creature they see after hatching as their mother
- The shared secret treated like a soul, the device is the body
- Model change of ownership as death followed by birth
  - Old key is discarded
  - Device readied for new imprinting
  - Reverse metempsychosis.— process of body inhabited by a succession of souls (like posession)
- What if owner loses control?
  - e.g. universal remote control broken?

- Escrowed Seppuku — Cause device to commit "suicide" using escrowed keys
- Alternatively Mother can back up the key
- Or Mother can partition the key into shares and distribute (using Rabin's approach)

Must I always use weak encryption?

- No, during imprinting much of the hard work can be done on a more powerful node (key generation and signing)
- During connection establishment use strong encryption to configure the connection (which may mix strong and weak encryption)

Integrity

Integrity means ensuring that the node has not been maliciously altered.

If you cannot afford signatures

- Checking the calibration of the devices is hard
- Use soft state to ensure that certificates expire
- Tamper resistence is not practical.
- Big Stick Priciple - Whoever physically controls the device can take it over.

What if Master delegates authority to a machine?

- Permanent master could just share private key
- But this has risks and can compromise security
- Need a mechanism for temporary delegation of authority
- So Parent can delegate authority and instruct duckling with a policy for backing up.

In some cases we might want ducklings to die of loneliness

- I.e. if they don't receive reinforcement from siblings they shut down

Cocaine Auction Protocol

Scenario - Auction where buyers and seller wish to maintain anonymity

- Seller should only be able to identify person making the winning bid after committing to the sale.
- The bidders should not be able to identify each other and the identity of the winning bidder should not be disclosed.

How does an auction work

- There are i rounds
- At the start of each round the seller announces bid price b_i.
- Each bidder has Δt seconds to respond to the bid
- As soon as one buyer says yes, he wins the round and becomes winner of round i, w_i
- If noone bids at round i the winner of the previous round w_{i-1} wins the auction.

Cryptographic Details

Before the start of the protocol all bidders and the seller agree on a system wide generating value g and a modulus n for a Diffie-Hellman model.

- It is widely believed that computing discrete logarithms is hard.
  - computing $g^x \mod n$ is believed much easier than estimating x given g and n
  - Is called a one-way function.
- However this is not known to be true.

The Protocol 1 of 2

The auction proceeds in rounds, i denotes the current round

- Each round has a fixed duration (say Δt)
  - The seller broadcasts
    - The bid price for the round b_i
    - The yes message of the winner of the previous round (if $i > 1$) $g^{x_{i-1}} \mod n$.
  - Each buyer, say buyer j, that is bidding in round i
    - Computes a nonce $x_{j,i}$ this is a private key
    - Generates a yes message, consisting of a public key $g^{x_{j,i}} \mod n$ and send it to the seller
    - Buyers not bidding are silent
  - At the end of a round of bidding, the seller counts the number of bids
    - Multiple yes bids - pick an arbitrary bid (e.g. the first), pick it as the winner of this round w_i.
    - Only one bid - it wins the round and the auction and is from w_i
    - No bids: if $i = 0$ no sale (wouldn't meet minimum price), else w_{i-1} wins the auction.

The Protocol 2 of 2

At the end of last round of bidding

- The seller remembers the public key of w_f, denoted $g^{x_f} \mod n$, winner's private key is x_f.
- The seller computes a nonce y (private key)
- The seller broadcasts $g^y \mod n$ (public key)
- The buyer AND seller now us $g^{x_f y} \mod n$ as their session key
- The seller broadcasts the transaction details encrypted with the session key $g^{x_f y} \mod n$

Bibliography