Ad Hoc (Wireless)
Key Establishment

Problem Definition

- **Goal:** Secure, authenticated communication between devices that share no prior context
- No prior context:
  - – No CAs or other trusted authorities
  - – No PKI
  - – No shared secrets
  - – No common history
- Problem: key establishment

- Diffie-Hellman shows how to share secrets…

Diffie-Hellman Key Agreement

- Public values: large prime p, generator g
- Alice has secret value a, Bob has secret b
- A → B: g^a mod p
- B → A: g^b mod p
- Bob: $(g^a \bmod p)^b \bmod p = g^{ab} \bmod p$
- Alice: $(g^b \bmod p)^a \bmod p = g^{ab} \bmod p$
- Eve cannot compute $\mathbf{g^{ab}}$ **mod p**

Are we done?

Problem: Man-in-the-middle Attack

- Mallory can impersonate Alice to Bob, and impersonate Bob to Alice!
  - A → M: g^a mod p
  - M → A: g^m mod p
  - M → B: g^m mod p
  - B → M: g^b mod p
  - Bob: $(g^m \bmod p)^b \bmod p = \mathbf{g^{bm} \bmod p}$
  - Alice: $(g^m \bmod p)^a \bmod p = \mathbf{g^{am} \bmod p}$

How Serious is MitM Attack?

- Wireless communication is invisible
  - – People can’t tell which devices are connected
- Neighbor can easily execute MitM attack
  - – If neighbor has a faster computer, it can easily respond faster than the legitimate devices
- **Easy to perform with high success rate!**

Solution?

Solution to Man-in-the-Middle Attack

- **Authentication!**

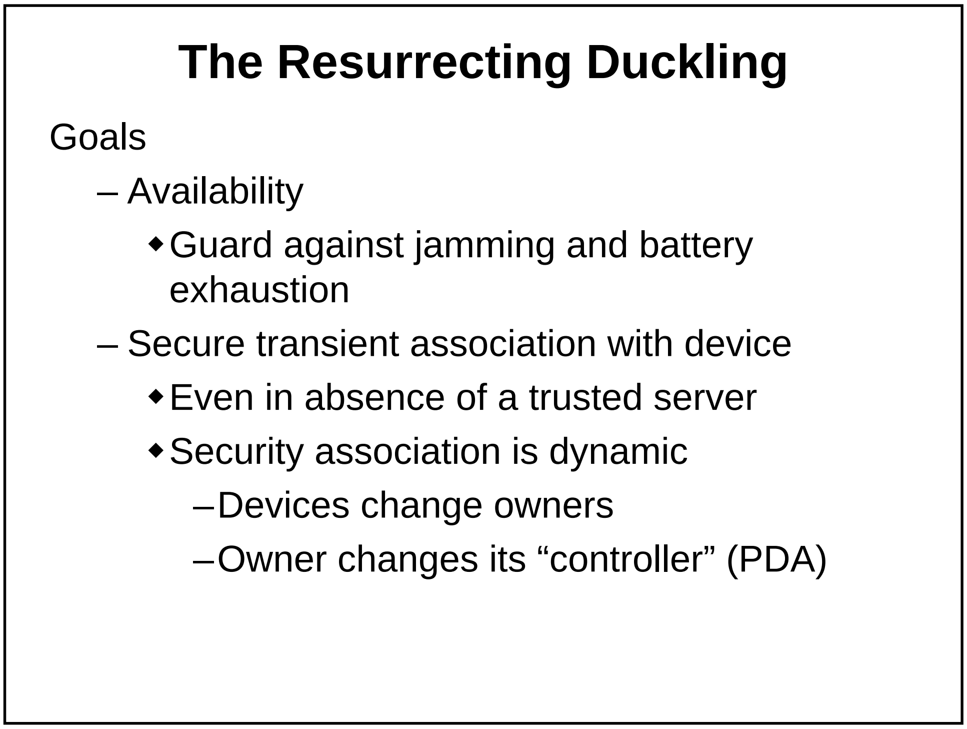
- Public DH values **must be authenticated**
- How?
  - – Tradeoffs between **security**, **usability**, and **transparency** to the user
  - – Transparency:
    - ◆ Does the user **realize** s/he is involved in a key establishment protocol?
    - ◆ Does the user **need** to realize this?

Resurrecting Duckling

- F. Stajano and R. Anderson, IWSP ’99
- **Problem:** how can we set up keys in a ubiquitous computing environment?
  - Devices use wireless communication
  - Setup keys between household devices and a PDA
- **Solution?**

The Resurrecting Duckling

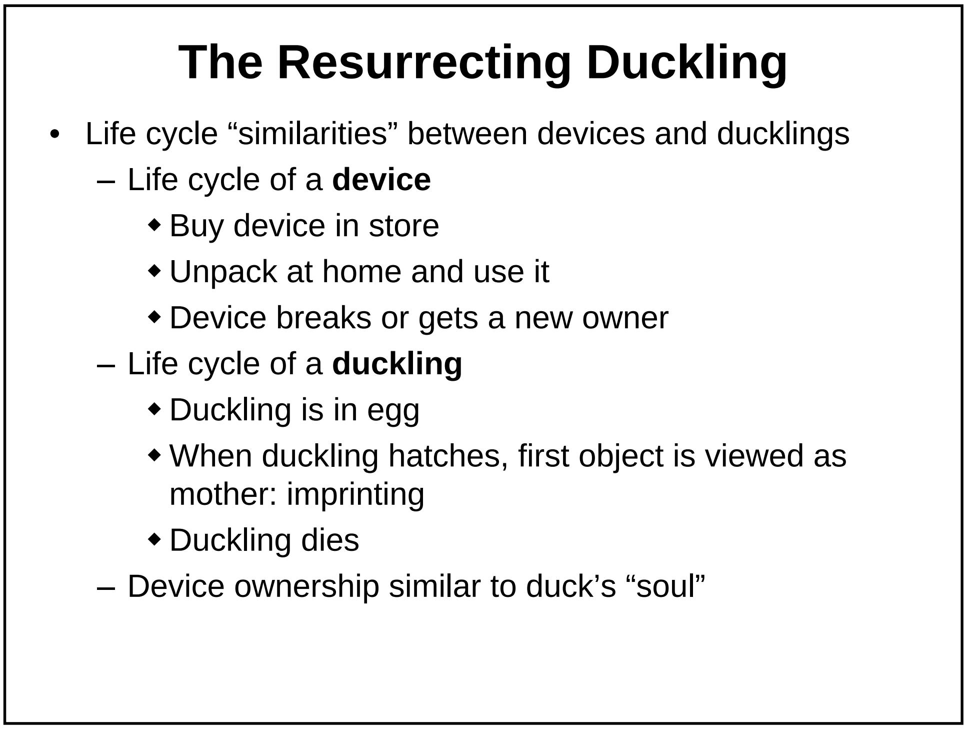
- **Solution:** set up keys using **trusted communication channel**
  - No cryptographic keys to setup this channel
  - Physical (WIRED) contact establishes a secure channel



The Resurrecting Duckling

Goals

- Availability
  - Guard against jamming and battery exhaustion
- Secure transient association with device
  - Even in absence of a trusted server
  - Security association is dynamic
    - Devices change owners
    - Owner changes its “controller” (PDA)



The Resurrecting Duckling

- Life cycle “similarities” between devices and ducklings
  - Life cycle of a **device**
    - Buy device in store
    - Unpack at home and use it
    - Device breaks or gets a new owner
  - Life cycle of a **duckling**
    - Duckling is in egg
    - When duckling hatches, first object is viewed as mother: imprinting
    - Duckling dies
  - Device ownership similar to duck’s “soul”

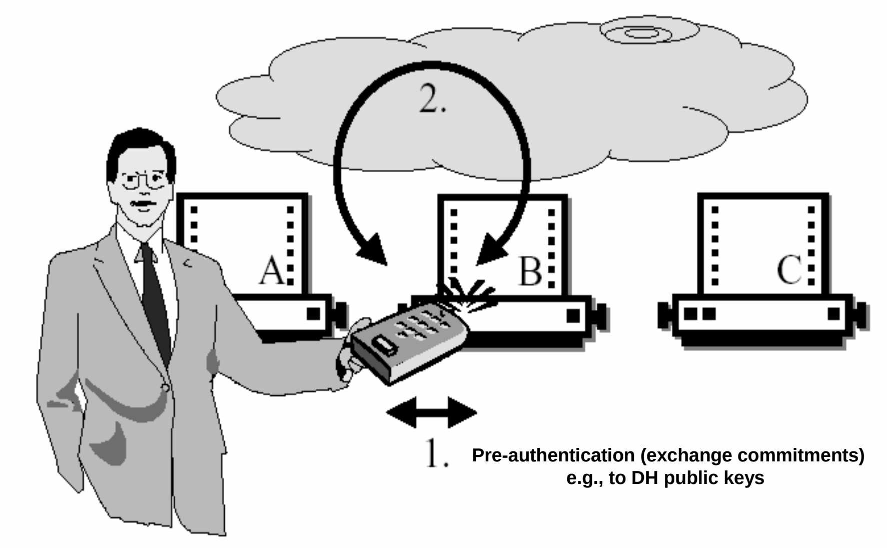
The Resurrecting Duckling

- Device life cycle
  - Device **imprinted** by master when it wakes up
  - Reincarnation:
    - ◆Device dies and gets new owner
  - Escrowed suicide:
    - ◆Manufacturer can “kill” device to enable renewed imprinting
- Physical contact establishes secure key during imprinting phase
  - MitM ‘impossible’ over physical contact channel
  - Diffie-Hellman can be safely performed

Talking to Strangers

- Balfanz et al. NDSS ’02
- Addresses practical shortcomings of Duckling
  - Devices have no interfaces for physical contact
  - Cables are cumbersome
- Propose Infrared as a “**Location-Limited Side Channel”**
  - Assumed to be immune to MitM attack
  - Many of today’s devices equipped with IR
  - Want **demonstrative identification** of devices

Talking to Strangers



Pre-authentication (exchange commitments)
e.g., to DH public keys

Talking to Strangers

- Pros
  - Works on many commodity devices
- Cons
  - Most users do not know where their IR port is
  - IR is invisible, attacker may still be able to mount MitM attack
  - Demonstrative identification achieved only if IR works correctly

Key Agreement in P2P Wireless Networks

- M. Cagalj, et al.
  Proc. of IEEE, Special Issue on Security and Cryptography, ‘05

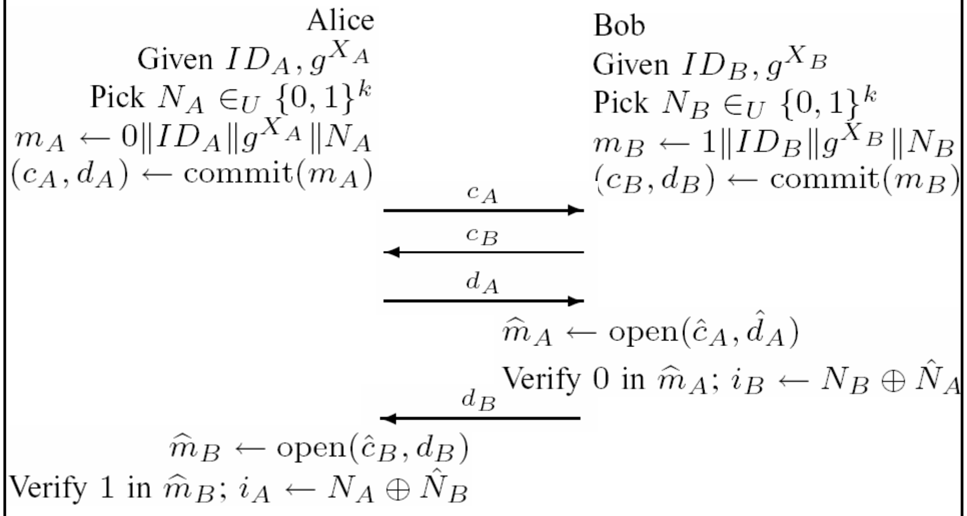
- Avoids use of side-channels
- Uses Diffie-Hellman to establish keys
- Presents three techniques to combat MitM
  - Visual comparison of short strings
  - Distance bounding
  - Integrity codes
- All 3 authenticate public DH parameters g^A and g^B

Commitment Schemes

- All 3 techniques use commitment schemes
- Commitment semantics:
  - Binding
  - Hiding
- $(c, d) \leftarrow commit(m)$
- m – message
- c – commitment value
- d – opening value

- It is infeasible to find *d’* such that *(c, d’)* reveals *m’ ≠ m*

DH using Short String Comparison (DH-SC)



Alice

Given ID_A, g^{X_A}

Pick $N_A \in_U \{0,1\}^k$

$m_A \leftarrow 0\|ID_A\|g^{X_A}\|N_A$

$(c_A, d_A) \leftarrow \text{commit}(m_A)$

Bob

Given ID_B, g^{X_B}

Pick $N_B \in_U \{0,1\}^k$

$m_B \leftarrow 1\|ID_B\|g^{X_B}\|N_B$

$(c_B, d_B) \leftarrow \text{commit}(m_B)$

$\xrightarrow{c_A}$

$\xleftarrow{c_B}$

$\xrightarrow{d_A}$

$\widehat{m}_A \leftarrow \text{open}(\hat{c}_A, \hat{d}_A)$

Verify 0 in $\widehat{m}_A$; $i_B \leftarrow N_B \oplus \hat{N}_A$

$\xleftarrow{d_B}$

$\widehat{m}_B \leftarrow \text{open}(\hat{c}_B, d_B)$

Verify 1 in $\widehat{m}_B$; $i_A \leftarrow N_A \oplus \hat{N}_B$

Alice and Bob manually compare i_A and i_B

If $i_A = i_B$, Alice and Bob output “Accept” $\hat{m}_B$ and $\hat{m}_A$, respectively.

DH-SC Analysis

- Pros
  - – Can be parameterized with shorter strings
  - – Tradeoff between usability and security
- Cons
  - – Users manually compare i_A and i_B
  - – Requires user diligence

- Why use commitments? Why not just compare the hash of the public DH values?

DH-SC Analysis

- Why use commitments? Why not just compare the combined hash of the two public DH values?

- Attacker has control of inputs to **both** hash functions
- Short string greatly reduces search space for an attacker to find *collisions*
  - This is dangerous
  - Requires attack on strong collision-resistance of hash function
  - Recall recent results against MD5 and SHA-1

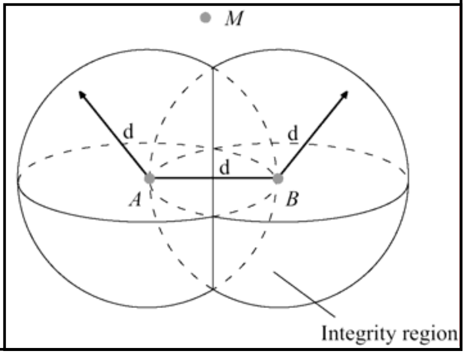
However, could perform two comparisons and forget the Commitments…

Reminder: Desired (cryptographic) Hash Function Properties

- Pre-image resistance (one-way-ness)
  - Given y = h(x) it is difficult to find x
- Second Pre-image resistance
  - A.k.a. “weak” collision resistance
  - For a given x, it is difficult to find x’ such that h(x) = h(x’)
  - Attacker chooses only **one** input
  - Used in digital signatures
- Collision resistance
  - A.k.a. “strong” collision resistance
  - It is difficult to find x and x’ such that h(x) = h(x’)
  - Attacker chooses **both** inputs

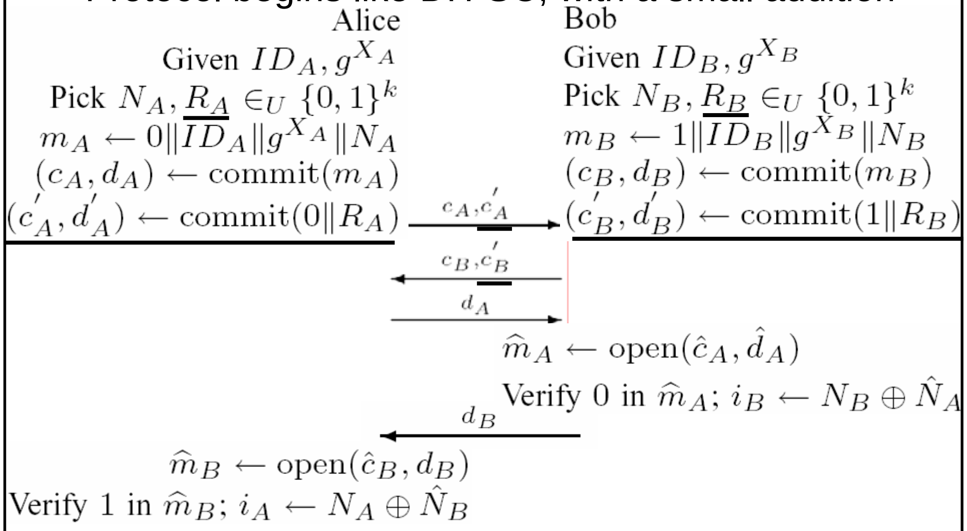
DH using Distance Bounding (DH-DB)

- Using precise timing by the radio interface, one can bound the maximum possible distance between devices *A* and *B*
- Results in an **integrity region** which provides proximity verification
- If users can visually verify there are no other users / devices within the integrity region, then $i_A = i_B$
- How does this work?



DH using Distance Bounding (DH-DB)

- Protocol begins like DH-SC, with a small addition



Alice

Given ID_A, g^{X_A}

Pick $N_A, \underline{R_A} \in_U \{0,1\}^k$

$m_A \leftarrow 0 \| \overline{ID}_A \| g^{X_A} \| N_A$

$(c_A, d_A) \leftarrow \text{commit}(m_A)$

$(c'_A, d'_A) \leftarrow \text{commit}(0 \| R_A)$

Bob

Given ID_B, g^{X_B}

Pick $N_B, \underline{R_B} \in_U \{0,1\}^k$

$m_B \leftarrow 1 \| \overline{ID}_B \| g^{X_B} \| N_B$

$(c_B, d_B) \leftarrow \text{commit}(m_B)$

$(c'_B, d'_B) \leftarrow \text{commit}(1 \| R_B)$

$\xrightarrow{c_A, \underline{c'_A}}$

$\xleftarrow{c_B, \underline{c'_B}}$

$\xrightarrow{d_A}$

$\widehat{m}_A \leftarrow \text{open}(\hat{c}_A, \hat{d}_A)$

Verify 0 in $\widehat{m}_A$; $i_B \leftarrow N_B \oplus \hat{N}_A$

$\xleftarrow{d_B}$

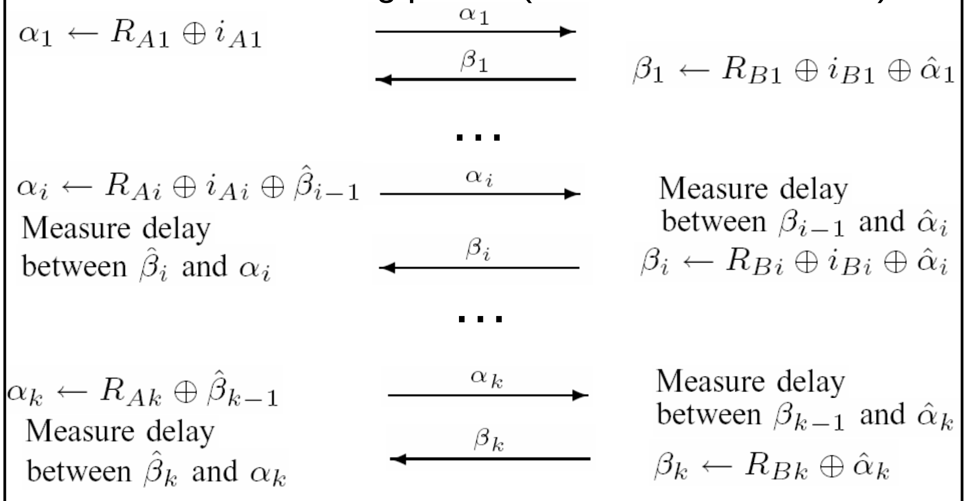
$\widehat{m}_B \leftarrow \text{open}(\hat{c}_B, d_B)$

Verify 1 in $\widehat{m}_B$; $i_A \leftarrow N_A \oplus \hat{N}_B$

- Next, we use distance-bounding to verify $i_A = i_B$

DH using Distance Bounding (DH-DB)

- Distance-bounding phase (Brands & Chaum '93)



$\alpha_1 \leftarrow R_{A1} \oplus i_{A1}$ $\xrightarrow{\alpha_1}$

$\xleftarrow{\beta_1}$ $\beta_1 \leftarrow R_{B1} \oplus i_{B1} \oplus \hat{\alpha}_1$

. . .

$\alpha_i \leftarrow R_{Ai} \oplus i_{Ai} \oplus \hat{\beta}_{i-1}$ $\xrightarrow{\alpha_i}$ Measure delay between β_{i-1} and $\hat{\alpha}_i$

Measure delay between $\hat{\beta}_i$ and α_i $\xleftarrow{\beta_i}$ $\beta_i \leftarrow R_{Bi} \oplus i_{Bi} \oplus \hat{\alpha}_i$

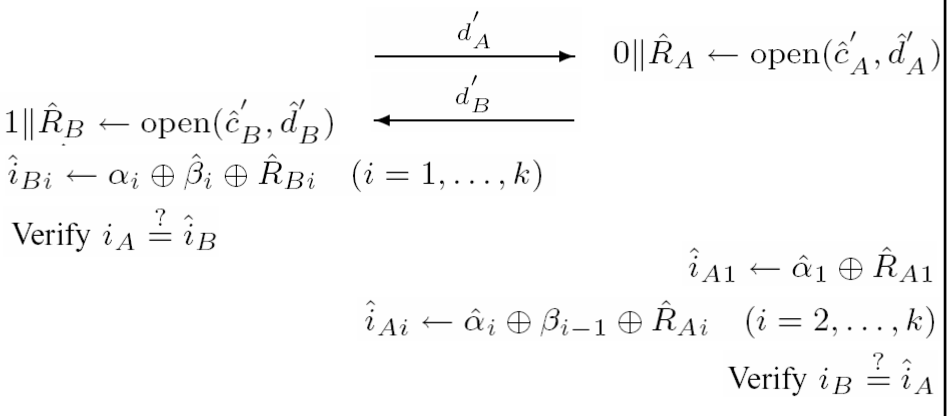
. . .

$\alpha_k \leftarrow R_{Ak} \oplus \hat{\beta}_{k-1}$ $\xrightarrow{\alpha_k}$ Measure delay between β_{k-1} and $\hat{\alpha}_k$

Measure delay between $\hat{\beta}_k$ and α_k $\xleftarrow{\beta_k}$ $\beta_k \leftarrow R_{Bk} \oplus \hat{\alpha}_k$

DH using Distance Bounding (DH-DB)

- End of distance bounding phase



$\xrightarrow{d'_A}$ $0 \| \hat{R}_A \leftarrow \text{open}(\hat{c}'_A, \hat{d}'_A)$

$1 \| \hat{R}_B \leftarrow \text{open}(\hat{c}'_B, \hat{d}'_B)$ $\xleftarrow{d'_B}$

$\hat{i}_{Bi} \leftarrow \alpha_i \oplus \hat{\beta}_i \oplus \hat{R}_{Bi} \quad (i = 1, \ldots, k)$

Verify $i_A \stackrel{?}{=} \hat{i}_B$

$\hat{i}_{A1} \leftarrow \hat{\alpha}_1 \oplus \hat{R}_{A1}$

$\hat{i}_{Ai} \leftarrow \hat{\alpha}_i \oplus \beta_{i-1} \oplus \hat{R}_{Ai} \quad (i = 2, \ldots, k)$

Verify $i_B \stackrel{?}{=} \hat{i}_A$

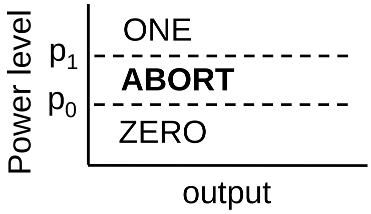
- Alice and Bob visually verify there are no other devices / users in their vicinity (the "integrity region")

DH-DB Analysis

- Pros:
  - i_A and i_B are compared by devices instead of users
  - Does not depend on the power ranges of devices
    - Depends solely on their proximity
  - Ultrasound requires millisecond timing precision
- Cons:
  - Pure RF implementation requires nanosecond timing precision (of XOR ops as well as radio)
    - To date, only Ultra Wide Band (UWB) can do this
    - Not available in commodity devices
  - Ultrasound available today, but not in commodity devices
  - No interference from other sw on devices…

DH using Integrity Codes (DH-IC)

- The sending radio transmits at only 2 power levels
  - Power level 0 indicates a logical 0
  - Power level p indicates a logical 1
- The receiver applies 2 thresholds (p_0 and p_1)
  - Signals above p_1 are a logical 1
  - Signals below p_0 are a logical 0
  - Signals between p_0 and p_1 **abort** the protocol



DH using Integrity Codes (DH-IC)

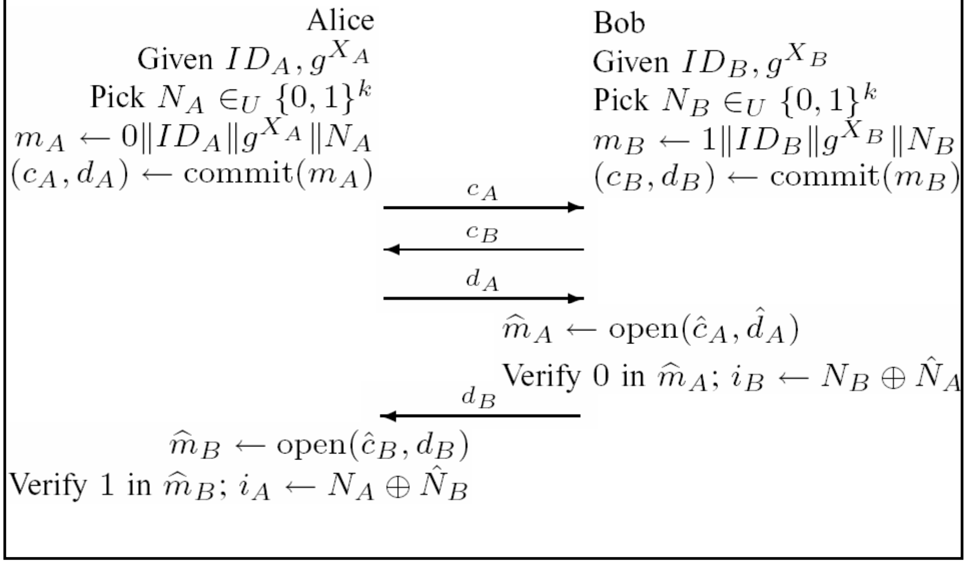
- Transmit messages in code words with a fixed number of 1’s
- Attacker can inject 1’s, but cannot remove 1’s
- The receiver must be turned on and listening on the correct channel during the sender’s transmission

- Example:

Messages:	00	01	10	11
Code words:	0001	0010	0100	1000

DH using Integrity Codes (DH-IC)

- Protocol begins as in DH-SC



Alice

Given ID_A, g^{X_A}

Pick $N_A \in_U \{0,1\}^k$

$m_A \leftarrow 0 \| ID_A \| g^{X_A} \| N_A$

$(c_A, d_A) \leftarrow \text{commit}(m_A)$

Bob

Given ID_B, g^{X_B}

Pick $N_B \in_U \{0,1\}^k$

$m_B \leftarrow 1 \| ID_B \| g^{X_B} \| N_B$

$(c_B, d_B) \leftarrow \text{commit}(m_B)$

c_A

c_B

d_A

$\widehat{m}_A \leftarrow \text{open}(\hat{c}_A, \hat{d}_A)$

Verify 0 in $\widehat{m}_A$; $i_B \leftarrow N_B \oplus \hat{N}_A$

d_B

$\widehat{m}_B \leftarrow \text{open}(\hat{c}_B, d_B)$

Verify 1 in $\widehat{m}_B$; $i_A \leftarrow N_A \oplus \hat{N}_B$

DH using Integrity Codes (DH-IC)

- Alice makes sure that Bob’s device is listening
- Alice pushes a button
- I-codes(i_A) sent to Bob’s device
- Alice announces “Message Sent” to Bob
- Bob updates his device (pushes a button)
- Verify I-code message integrity and $i_A \overset{?}{=} N_B \oplus \hat{N}_A$
- If verification okay, Alice and Bob output “Accept” $\hat{m}_B$ and $\hat{m}_A$, respectively

DH-IC Analysis

- User requirements
  - Alice must make sure Bob’s device is listening before pressing a button on her device
  - Bob then presses a button on his device
- Radio system requirements
  - It is not possible to block emitted signals without being detected, except with negligible probability
  - Multiple waveforms to send a ‘1’
  - No rigorous treatment of its feasibility

N. Asokan and P. Ginzboorg, “Key Agreement in Ad-hoc Networks,” *Computer Communications*, vol. 23, no. 17, pp. 1627–1637, 2000.

- Problem: how to set up a session key between a group of people/devices their who meet and have no prior context
- Shared password approach
- No PKI, no TTP
- Fresh password is chosen and manually shared among those present in the room (e.g., by writing on blackboard)
- Password used to derive a strong shared session key using either group DH or group-EKE
- Requires each user to type in the password

FYI: See paper on keyboard snooping from S&P’04

Seeing-is-Believing (SiB)

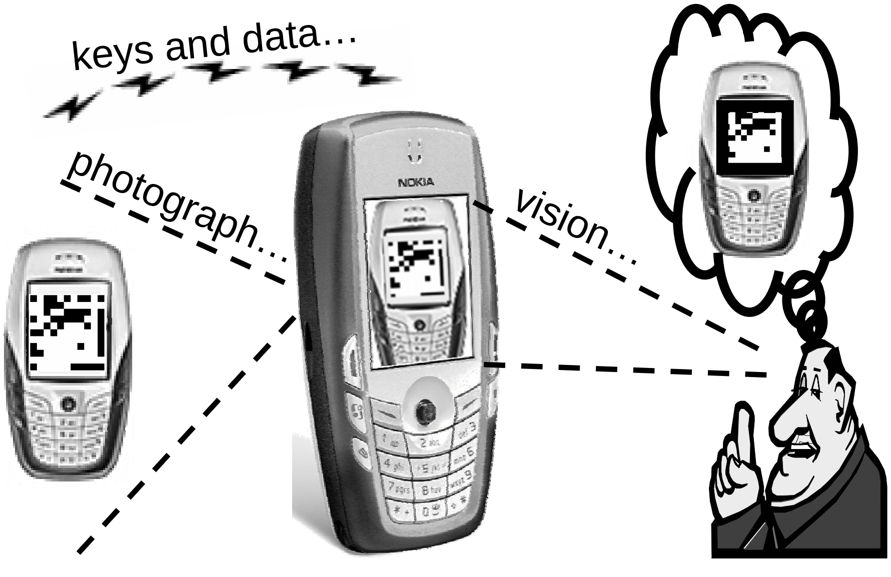
McCune et al. IEEE Security &Privacy ’05

- Difficult to achieve **demonstrative identification** of devices communicating wirelessly with no prior context
- Prior work proposes the use of a **location-limited side-channel** to authenticate devices
  - Infrared, ultrasound, physical contact
- Proposals to-date too cumbersome for non-expert users
  - None of them convince the user that they are really communicating with ***the target*** device

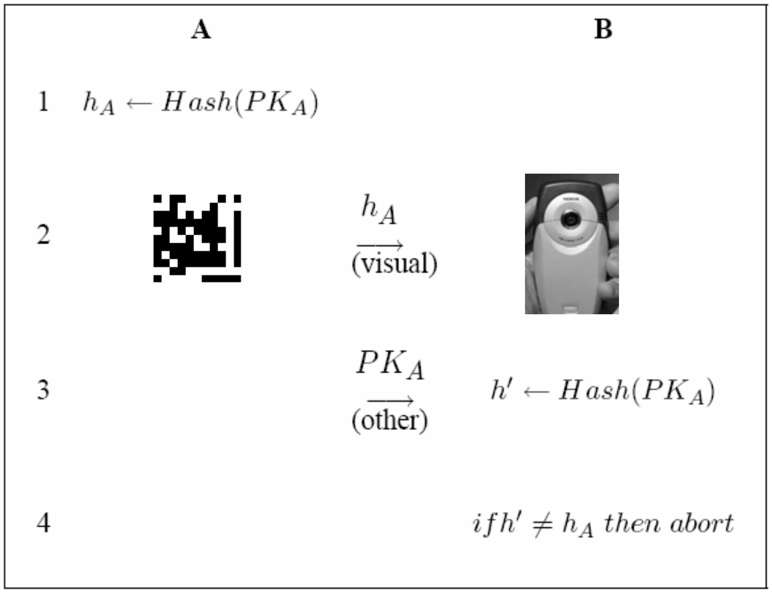
Seeing-Is-Believing

- Camera Phones now have sufficient resources to scan 2D barcodes
- Also have high-quality screens which can display freshly-generated barcodes
- Using them together yields a ***visual***, location-limited channel
- Visual channel ***can*** provide **demonstrative identification** of communicating parties to the user
- Enables strong authentication

Basic SiB Protocol



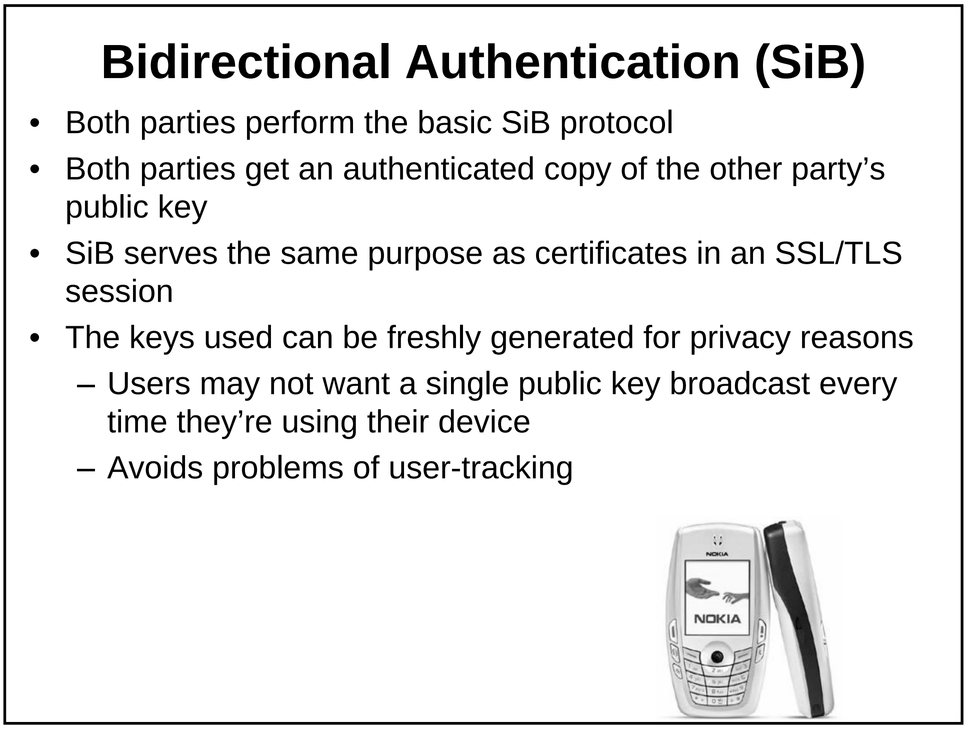
Basic SiB Protocol



	A		B
1	$h_A \leftarrow Hash(PK_A)$		
2		$\xrightarrow{h_A}$ (visual)	
3		$\xrightarrow{PK_A}$ (other)	$h' \leftarrow Hash(PK_A)$
4			$if h' \neq h_A$ $then\ abort$

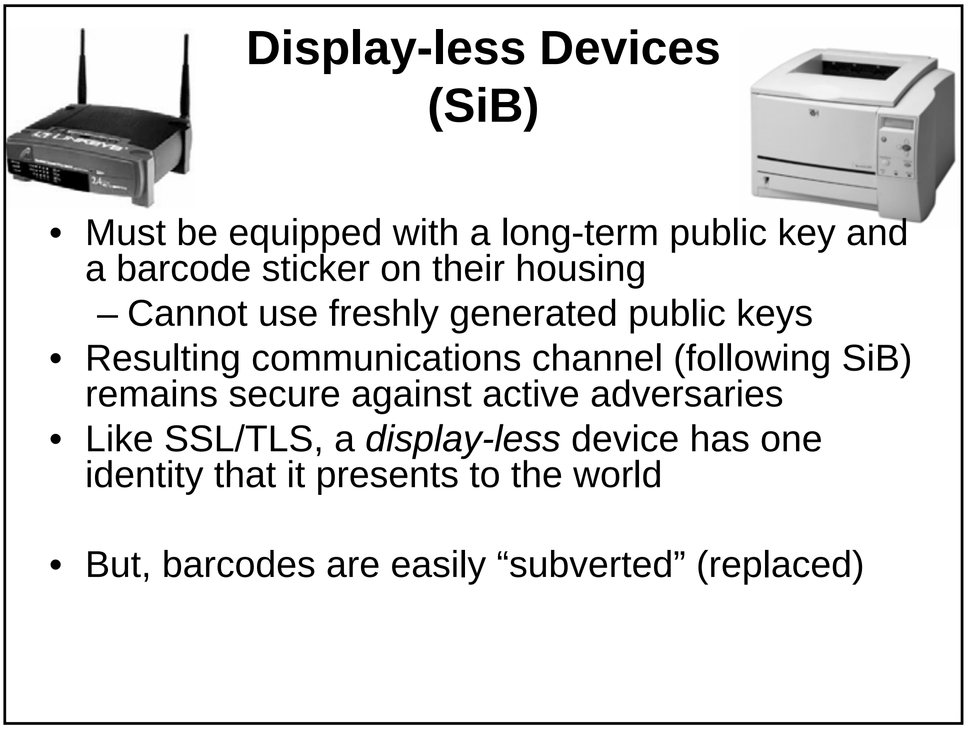
Device Configurations (SiB)

- Both devices have cameras and displays (most desirable configuration)
- SiB can be useful even if some devices are missing a camera, a display, or both
  - – Display-only **or** Camera-less
    - ◆Laptop, cable box, …
  - – Camera-less **and** Display-less
    - ◆802.11 access point, printer, …



Bidirectional Authentication (SiB)

- Both parties perform the basic SiB protocol
- Both parties get an authenticated copy of the other party's public key
- SiB serves the same purpose as certificates in an SSL/TLS session
- The keys used can be freshly generated for privacy reasons
  - Users may not want a single public key broadcast every time they're using their device
  - Avoids problems of user-tracking



Display-less Devices (SiB)

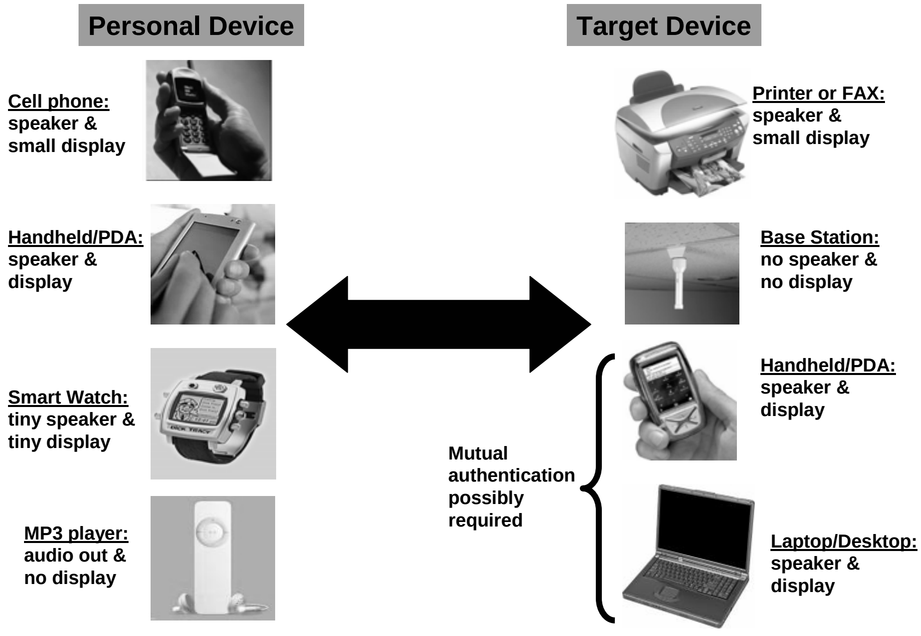
- Must be equipped with a long-term public key and a barcode sticker on their housing
  - Cannot use freshly generated public keys
- Resulting communications channel (following SiB) remains secure against active adversaries
- Like SSL/TLS, a *display-less* device has one identity that it presents to the world
- But, barcodes are easily "subverted" (replaced)

Loud and Clear (L&C) Security

M. Goodrich, et al. 2005

What if:

- Visually impaired user
- Not enough ambient light
- No camera-equipped device
- Afraid of barcode stickers being replaced?



Personal Device

Cell phone: speaker & small display

Handheld/PDA: speaker & display

Smart Watch: tiny speaker & tiny display

MP3 player: audio out & no display

Target Device

Printer or FAX: speaker & small display

Base Station: no speaker & no display

Handheld/PDA: speaker & display

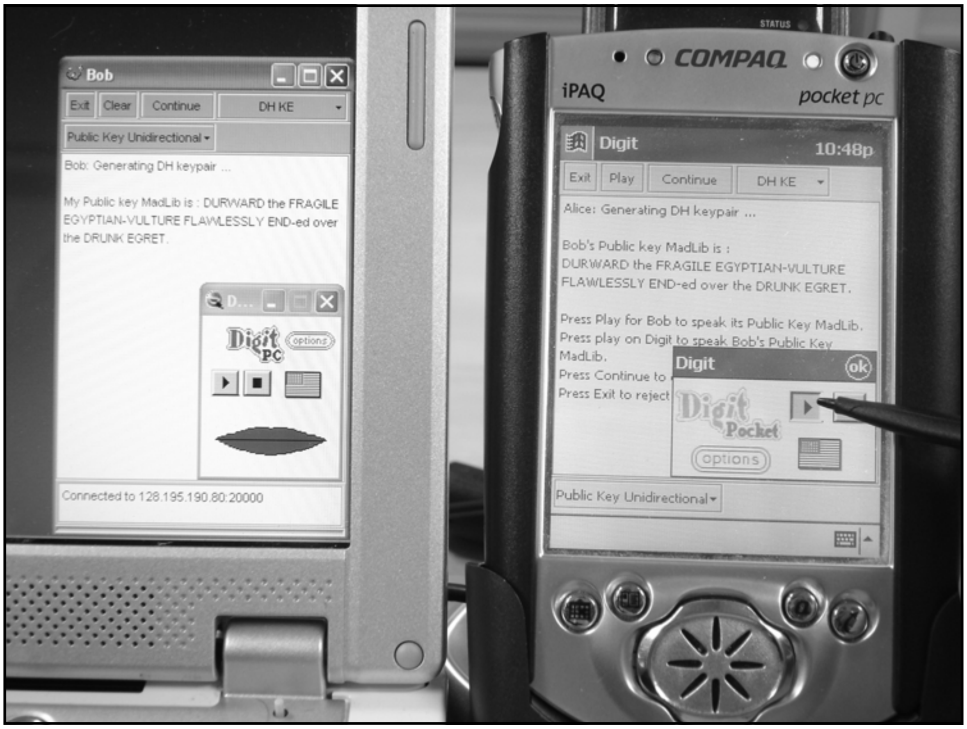
Laptop/Desktop: speaker & display

Mutual authentication possibly required

L&C Security

- Solution: use audio channel
- Human-assisted vocalized string comparison
- Exchange DH (or RSA) keys via any wireless (or wired) channel
- Hash other party's key and convert to MadLib sentence: non-sensical but grammatically-correct construction, e.g., 70-bit string represented as:

  DONALD the FORTUNATE BLUE-JAY FRAUDULENTLY CRUSH-ed over the CREEPY ARCTIC-TERN.



Bob: Generating DH keypair ...

My Public key MadLib is : DURWARD the FRAGILE EGYPTIAN-VULTURE FLAWLESSLY END-ed over the DRUNK EGRET.

Connected to 128.195.190.80:20000

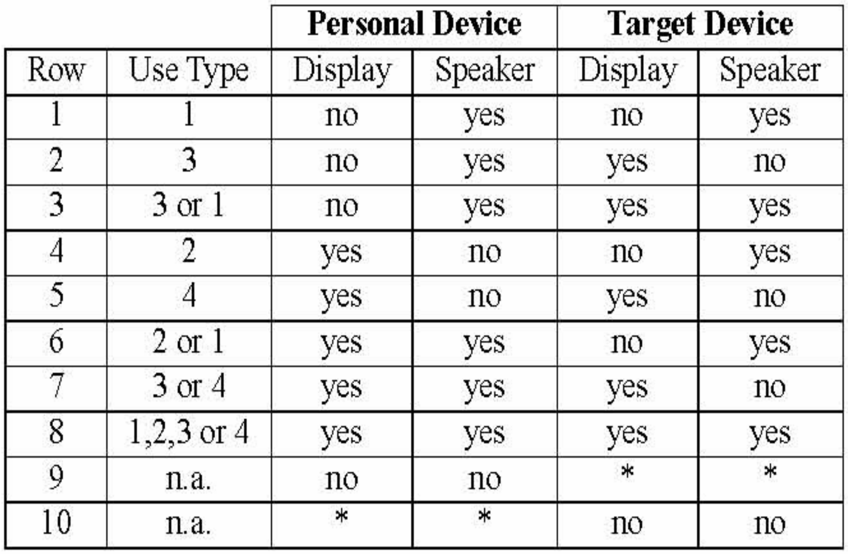
Alice: Generating DH keypair ...

Bob's Public key MadLib is :
DURWARD the FRAGILE EGYPTIAN-VULTURE FLAWLESSLY END-ed over the DRUNK EGRET.

Press Play for Bob to speak its Public Key MadLib.
Press play on Digit to speak Bob's Public Key MadLib.
Press Continue to
Press Exit to reject

Scenarios (use types):

- TYPE 1: hear and compare two audible sequences, one from each device.
- TYPE 2: hear an audible sequence from the target device and compare it to text displayed by the personal device.
- TYPE 3: hear an audible sequence from the personal device and compare it to text displayed by target device.
- TYPE 4: compare text displayed by the personal device to text displayed by target device.

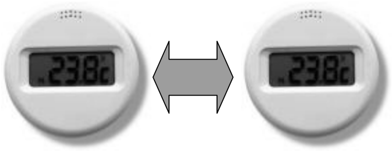


		Personal Device		Target Device	
Row	Use Type	Display	Speaker	Display	Speaker
1	1	no	yes	no	yes
2	3	no	yes	yes	no
3	3 or 1	no	yes	yes	yes
4	2	yes	no	no	yes
5	4	yes	no	yes	no
6	2 or 1	yes	yes	no	yes
7	3 or 4	yes	yes	yes	no
8	1,2,3 or 4	yes	yes	yes	yes
9	n.a.	no	no	*	*
10	n.a.	*	*	no	no

Shake Them UP

C. Castelluccia, P. Mutaf, Mobisys'05

- Need to pair (i.e., establish a shared secret on-the-fly) between two wireless devices
- Devices, such as sensors, have *very limited* CPU, memory and power!
- Standard methods such as the DH key exchange are not suitable
- Example:



Current Solutions

- PKC-based schemes
  - Rely on PK key exchange protocols such as RSA or DH
  - Perform CPU-intensive operations: modular exponentiations
  - Too expensive for tiny devices
- PIN-based schemes (for ex. Bluetooth)
  - Key derived from a PIN
  - PIN typically entered via out-of-band channel such as a keyboard.
  - Computationally efficient
  - …but requires a physical user interface (keyboard) …and most sensors do not have a keyboard ☹!
  - Security is pretty weak since it depends on the PIN….

Other Solutions (2)

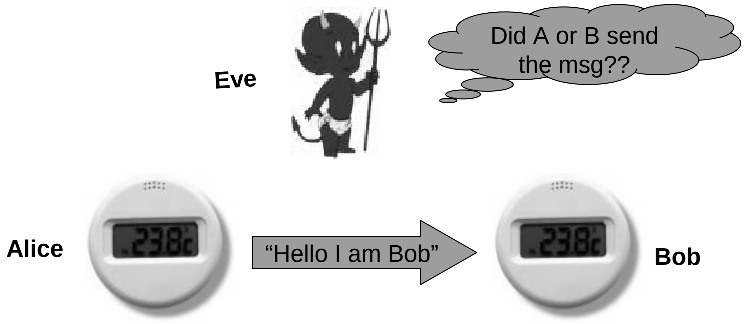
- Physical Contact (imprinting) - Duckling
  - establish a key via physical contact by linking devices with a wire….
  - not always practical and requires additional hardware..
- InfraRed channel - Strangers
  - IR is difficult to intercept since requires line-of-sight links.
  - most sensors do not have IR interface!
- Faraday Cage
  - Devices could be placed into a Faraday cage
  - It is clearly impractical to ask users to lug around a metal box ;-)

Goals

- Design a secure pairing protocols that:
  - Does not rely on PK cryptography
  - Does not rely on pre-configured information
  - Does not increase the complexity (and cost) of the sensors by requiring additional hardware such as a display, keyboard, IR channel…
  - Does not require special equipment (cable, faraday cage)
- Security Model
  - protocol must ensure that active or passive attackers do not learn the exchanged key
  - must provide some DoS protection,i.e. prevent an attacker from disrupting the key exchange and exhausting the devices' resources.

How to exchange one secret bit

- Let’s assume that Alice (A) and Bob (B) communicate over a wireless *anonymous* broadcast channel
  - Eve can read the exchanged packets
  - ...but can not identify the source of the packets.



Eve

Did A or B send the msg??

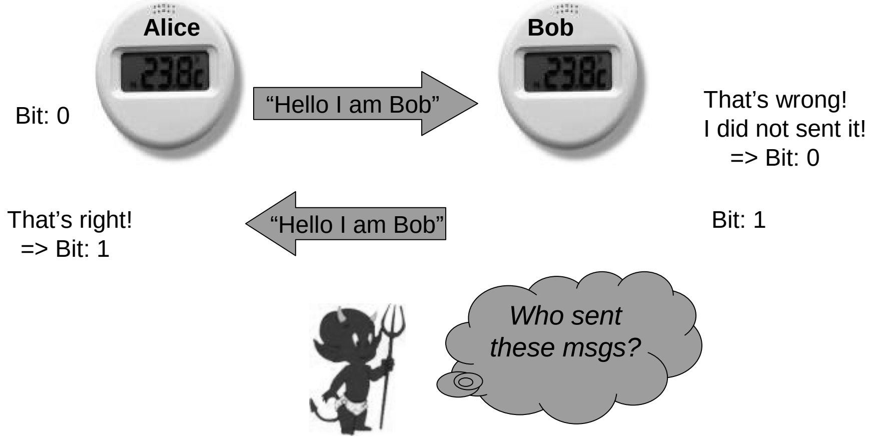
Alice

“Hello I am Bob”

Bob

How to exchange one secret bit (2)

- Alice and Bob can then use the following algorithm:



Alice

Bob

Bit: 0

“Hello I am Bob”

That’s wrong!
I did not sent it!
=> Bit: 0

That’s right!
=> Bit: 1

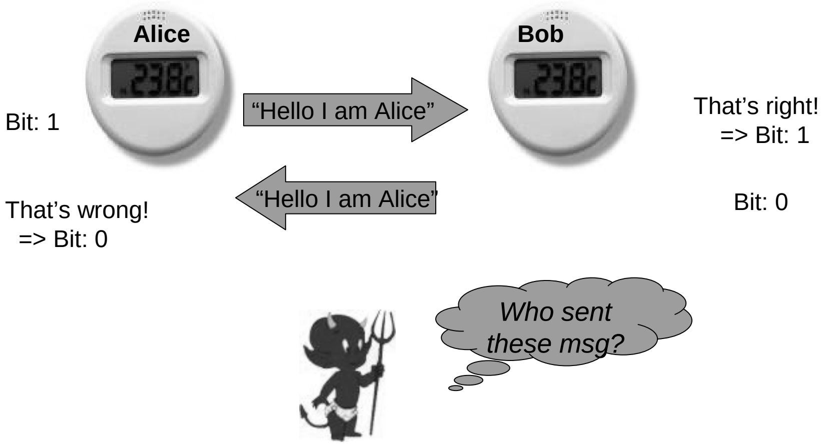
“Hello I am Bob”

Bit: 1

Who sent these msgs?

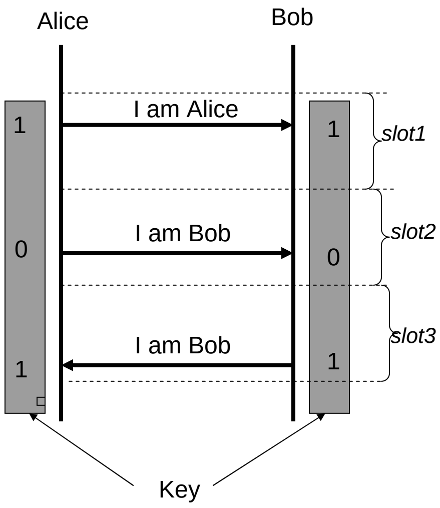
How to exchange one secret bit (3)

- Of course the protocol is symmetrical i.e. Alice can also send the bit “1” and Bob the bit “0”



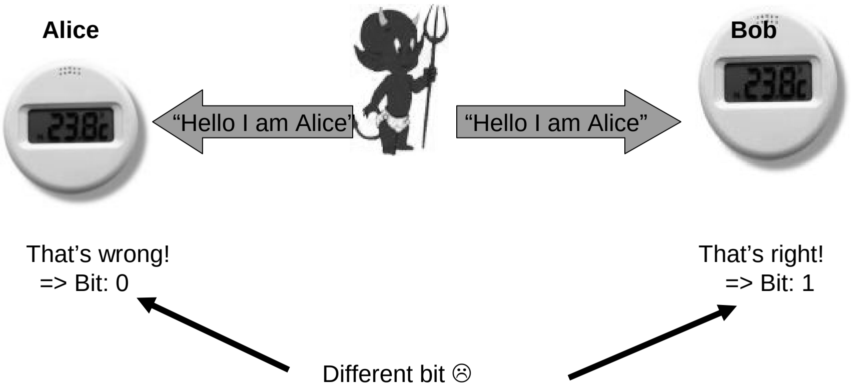
How to exchange N-bit secret

- Divide the time in N slots.
- In each slot, either A or B sends a message
- Transmission order is random ➔ Eve can not group the messages and retrieve the key…



Key poisoning Attack

- What if Eve injects a fake message?



Alice

Bob

“Hello I am Alice”

“Hello I am Alice”

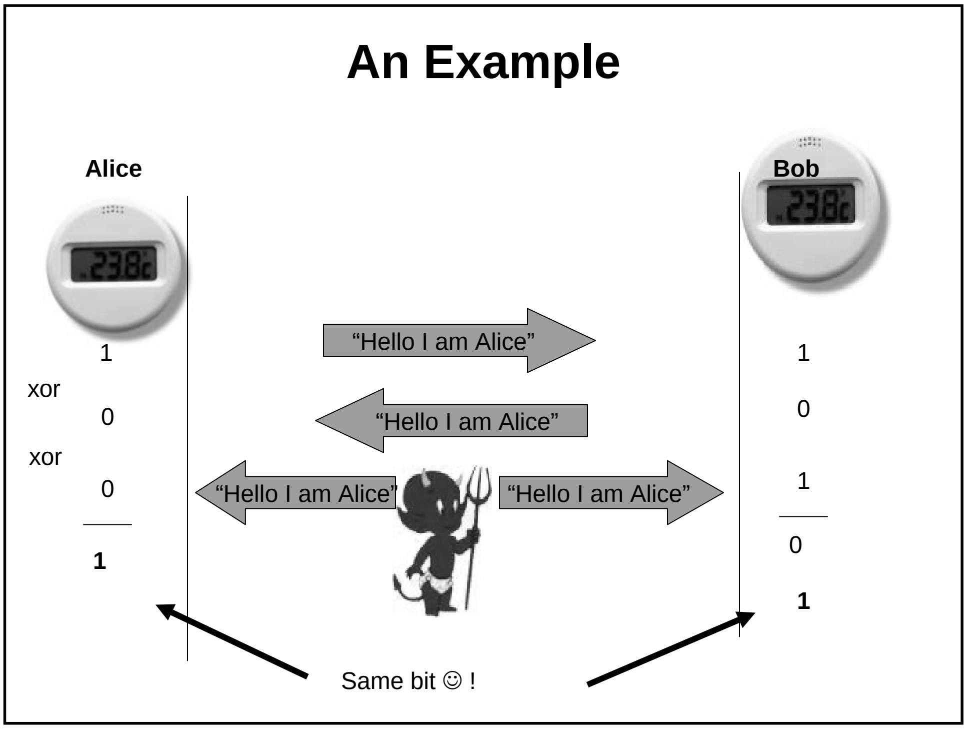
That’s wrong!
=> Bit: 0

That’s right!
=> Bit: 1

Different bit ☹

Key Poisoning Protection

- Both Alice and Bob must send one message during a specified time slot T at a *random* time in [0,T]
- Alice and Bob expect 2 messages per time slot
- If more than 2 packets are received …then there is a DoS attack!
- To compute the secret bit:
  - Alice XORs all received bits..
  - Bob XORs all received bits
    - if number of messages is odd it takes the inverse of the XORed bit



An Example

Alice

1

xor

0

xor

0

1

“Hello I am Alice”

“Hello I am Alice”

“Hello I am Alice”

“Hello I am Alice”

Bob

1

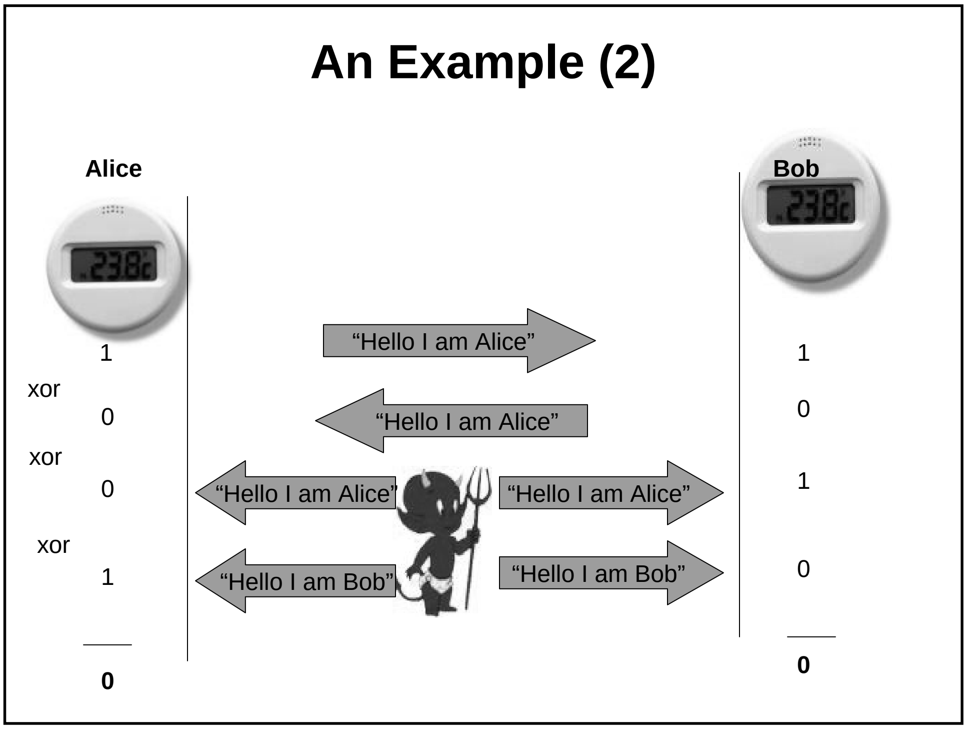
0

1

0

1

Same bit ☺ !



An Example (2)

Alice

1

xor

0

xor

0

xor

1

0

“Hello I am Alice”

“Hello I am Alice”

“Hello I am Alice”

“Hello I am Alice”

“Hello I am Bob”

“Hello I am Bob”

Bob

1

0

1

0

0

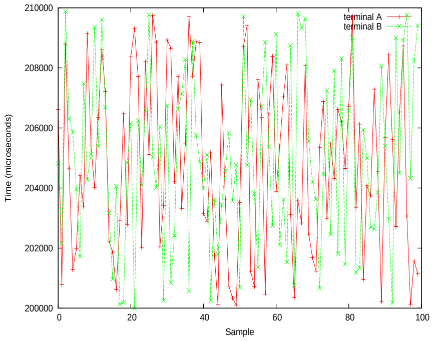
Wireless Anonymous Communication

- We assume source anonymity…
  - Can an 802.11-based system provide source anonymity?
- Eve can potentially identify the real source of the messages
  - Timing information
  - Reception Power
  - Frequency

Wireless Anonymous Communication (2)

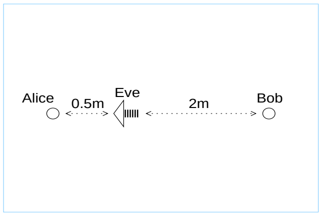
- Timing
  - This is quite trivial in TDMA based scheme since devices always transmit during their allocated slots
  - However Timing does not provide any information if a random access MAC protocol, such as CSMA, is used since each device access the channel at a random time!

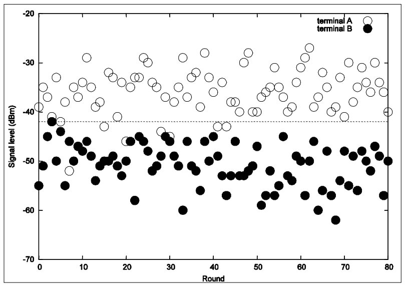
=> Protocol only works with CSMA-based technologies, such 802.11,802.15.4



Wireless Anonymous Communication (3)

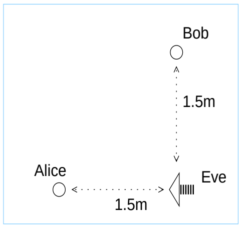
- Reception Power
  - If Eve is closer to Alice than Bob, she will receive Alice's message which a higher power!
  - Note: assume A and B transmit at the same power level.

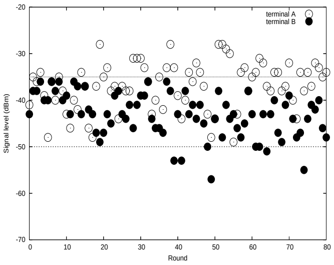




What can be done? (1)

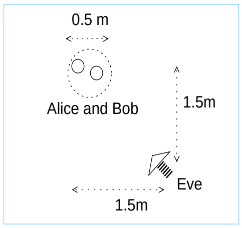
- Can randomly change Alice and Bob's transmission power
  - Some bits will still be revealed
  - If Eve has a directional antenna she can aim it at one of the devices!

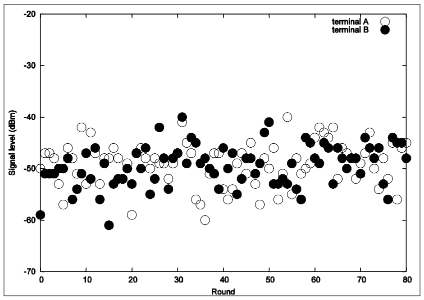




What can be done? (2)

- We can bring the devices together and move them (shake them up) one around the other!
  - The reception power of A’s and B’s messages will be similar…
  - Eve cannot use a directional antenna since the devices are moving!
- In summary, shaking 2 devices prevents using power to identify source!



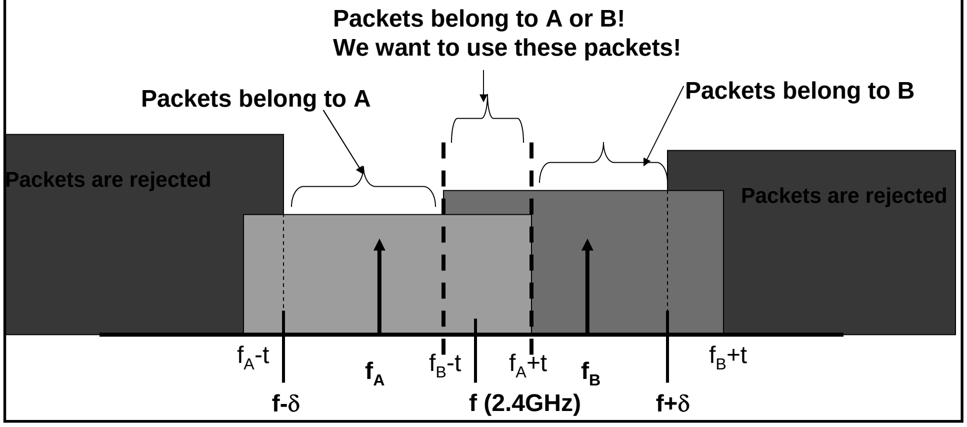


Frequency Fingerprinting

- Even though standard specify one frequency, each device uses a different frequency.
- Difference due to the crystal oscillator and clock drift, resulting from aging, temperature and so on.
- Typically an error of 25ppm (parts per million) is allowed
- So, if transmitting frequency is 2.4GHz, a frequency offset of up to 120kHz is allowed.
- Possibly, a (well-equipped) Eve can use this frequency difference to identify the source and retrieve the secret…

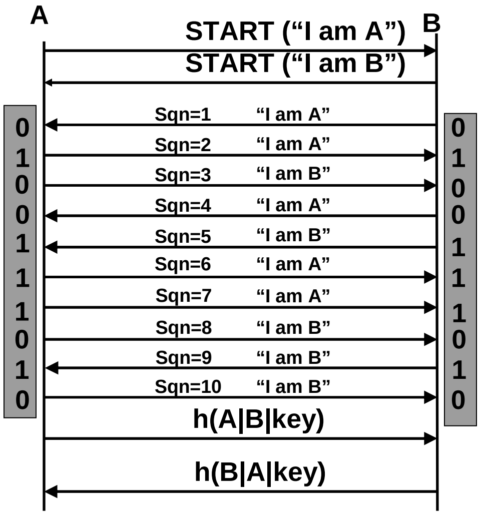
Frequency Fingerprinting (2)

- If you move the devices at a high speed, the doppler effect might solve the problem for you ☺ !
- A more practical solution is to add a random frequency offset so that A and B span over similar frequency ranges.
  - Btw This solution does not require modifying the standard!



The Shake' em Up protocol (STU)

- We combine the previous protocol with shaking.
- A user that wants to pair to devices A and B
  - Brings the devices together
  - Shakes them up!
  - Triggers the protocol (for example by pushing a bottom on the devices)…



Performance: Energy Consumption

- In STU, each device
  - processes N small messages, where N is # of bits of the secret (total number of bits sent: 2016)
  - …but performs almost no computation.
- In a DH-based scheme,
  - each node sends only one large message (>1024 bits)…
  - but performs a lot of computation…i.e. 4.12x10^{8} single precision multiplications (if N=72).
- By using the heuristic that transmitting one bit consumes as much energy as executing 800 instructions…
  - this scheme is ca. 100 times more energy efficient than a plain DH-based scheme

Conclusions

- Key establishment in ad hoc networks requires a trade-off between security, usability, and transparency to the user
- It is not necessarily desirable to have a totally human-transparent scheme
- If possible, involve the user, but in a way that is intuitive
- Taking pictures of desired communication endpoints is one way to achieve this property
- Listening is another way
- And shaking (juggling?) works too…