



Certified Wireless USB Association Models Supplement 1.0

Preston Hunt

Chair & Editor, Association Model Working Group
Intel Corporation

Presentation Objective



- **Review** why we need association
- Explain **requirements** for products
- Walk through **detailed examples** of cable and numeric models



What's Association?

- Scenario
 - Buy a new Certified Wireless USB digital camera
 - Want to connect to laptop for the first time
 - Easy, but also secure
- How?

Association Models Supplement defines how this is done in Certified Wireless USB

What's at Stake?



RFID Kills.com
Death by Passport.
 The State Department wants to turn all US passports into terrorist beacons.

SC
MAGAZINE
 for IT security professionals

- Home
- News
- Products
- Features

Go to **SC M**

News

Mobile virus infects Lexus cars
 by David Quainton

Lexus cars may be vulnerable to viruses that in mobile phones. Landcruiser 100 models LX470 :

Slashdot
 It is what IT is.

Car RFID Security System Cracked

 **tom's networking**

How To: Building a BlueSniper Rifle – Part 1

A Toronto Man faces charges after being arrested for "War-Driving" around Toronto neighborhoods, using unsecured networks to download child-pornography.



“Security” vs. “Association”

- These are split in the core spec
- Association
 - First time set-up of device to host
 - Host → Device: Connection context
 - Everything before 4-way handshake
- Security
 - Ongoing operational security (AES, MIC, ...)
 - Everything from 4-way handshake on
 - Be sure to attend Security presentation (today @ 5pm)



Association Model Specification

- Association Model Supplement 1.0
 - Definitive source for all association information
 - Full technical, implementation details
 - Everything you need to build devices
- Released as addendum to core spec
 - Covered by same rules as core spec
- Enforced through compliance testing and logo certification
- A few errata will be published soon

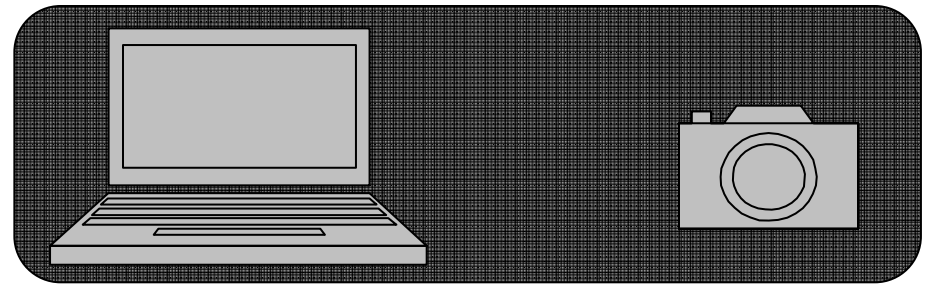


Association Model 1.0

- Two models: Cable and numeric
- Hosts **must** support cable and numeric
 - Limited hosts/DRDs need only support TPL list
- Devices with USB ports **must** support cable model
- Devices with displays **must** support numeric model
 - Diffie-Hellman also required
- Devices **must** use at least one of the above
- Will investigate NFC for future adoption (not in 1.0)



User Experience



You bring the device into the vicinity of the host.

You plug a cable into the host.

Then you plug the cable into the device.

The host will display a message that says:

“Success!

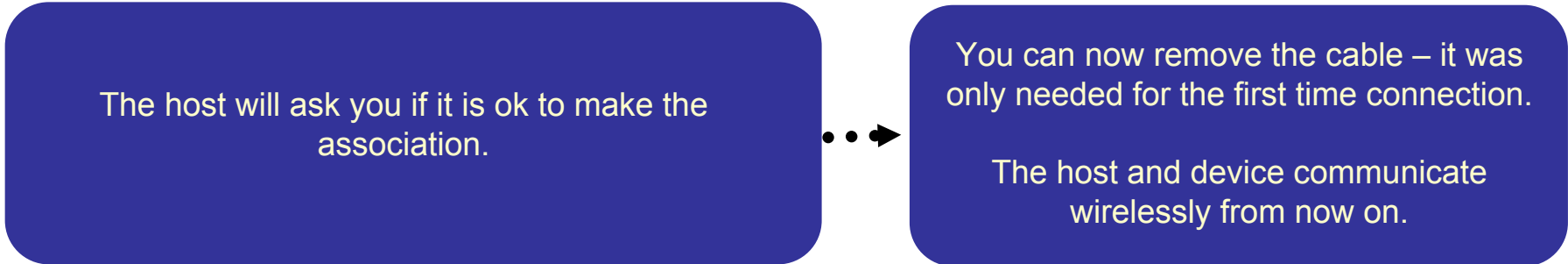
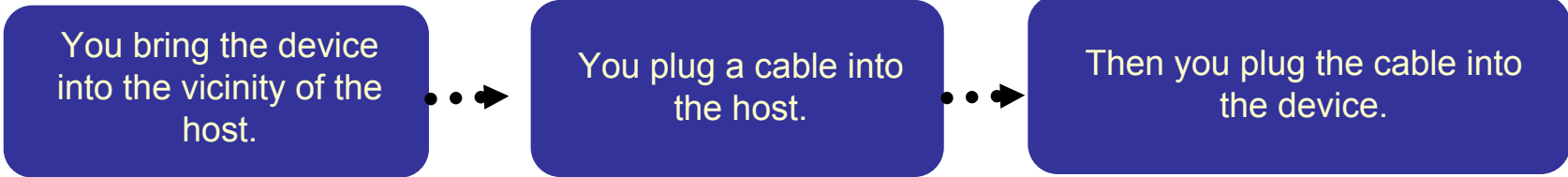
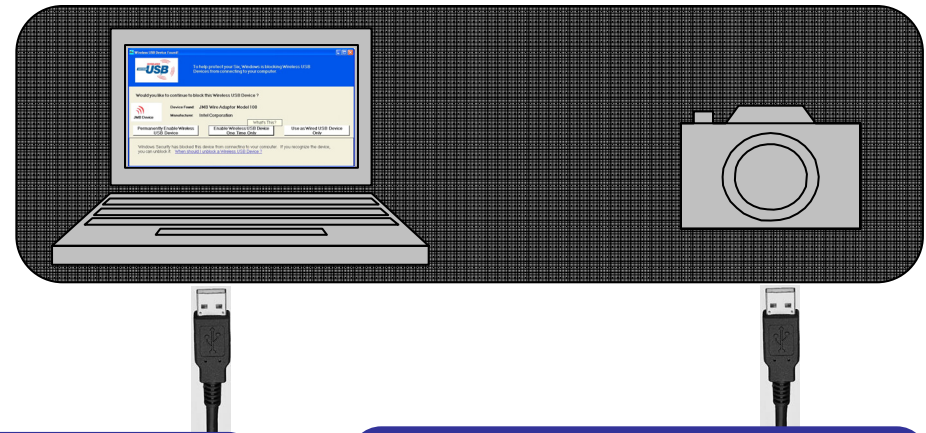
You may now use this device wirelessly.”

You can now remove the cable – it was only needed for the first time connection.

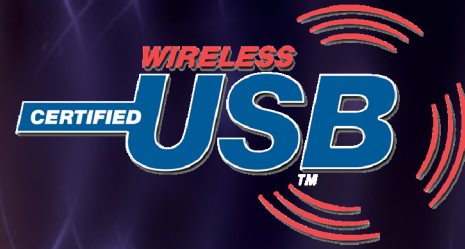
The host and device communicate wirelessly from now on.




User Experience with Confirmation



Hypothetical Confirmation UI




Wireless USB Device Found!



To help protect your Six, Windows is blocking Wireless USB Devices from connecting to your computer.

Would you like to continue to block this Wireless USB Device ?



Device Found: JMB Wire Adaptor Model 100
Manufacturer: Intel Corporation

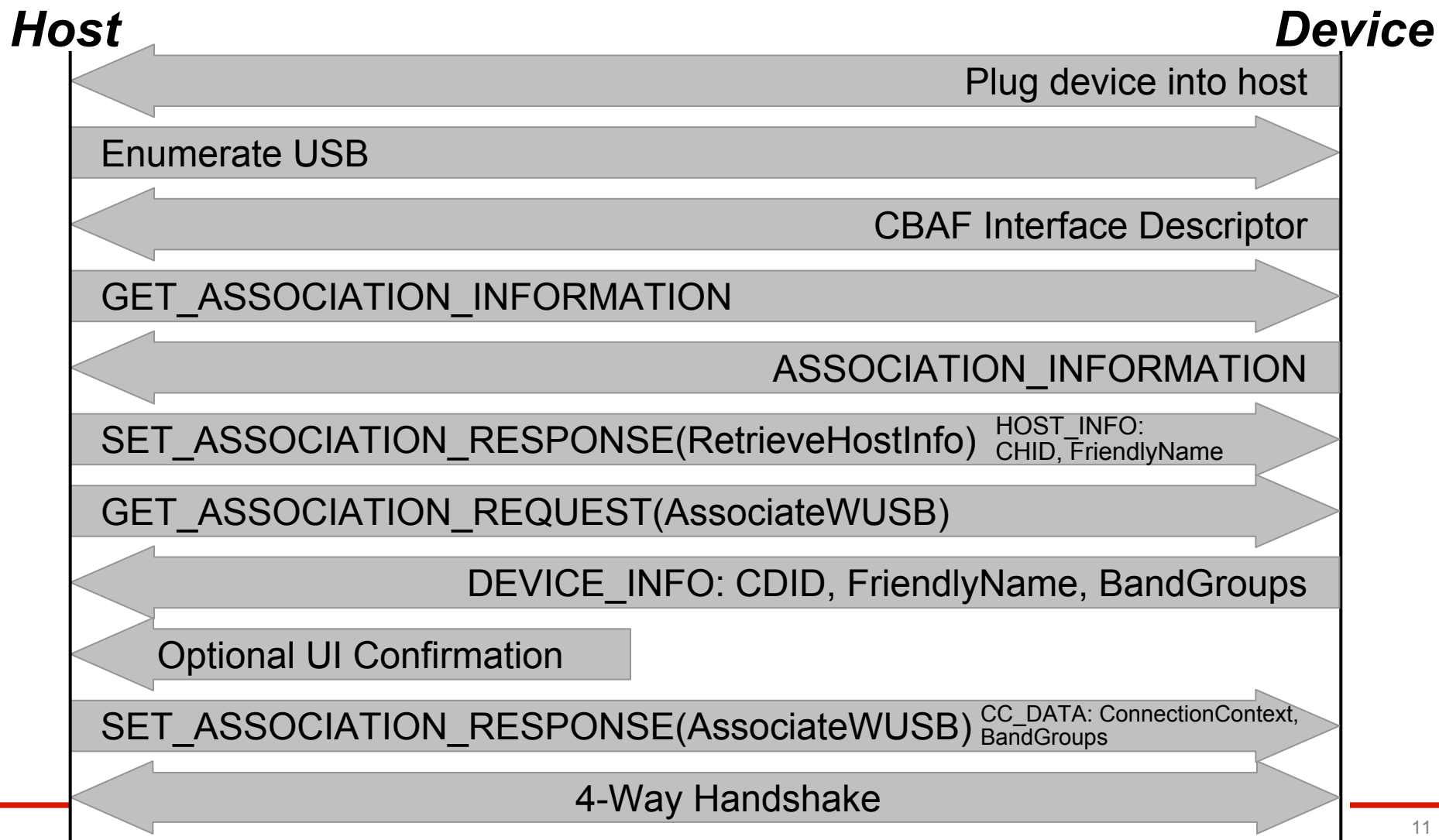
What's This?

Permanently Enable Wireless USB Device Enable Wireless USB Device One Time Only Use as Wired USB Device Only

Windows Security has blocked this device from connecting to your computer. If you recognize the device, you can unblock it. [When should I unblock a Wireless USB Device ?](#)



Cable Model Process - Preview





Detailed Walkthrough

- Cable-based Association Framework
 1. Reporting/Discovery
 2. Get Association Types
 3. RetrieveHostInfo
 4. AssociateWUSB
- Generic framework
 - Only will discuss Certified Wireless USB here
 - Could easily be expanded for others



Step 1: Reporting/Discovery

- Device adds CBAF Interface to its config descriptor set

bInterfaceClass	0xEF
bInterfaceSubClass	0x03
bInterfaceProtocol	0x01

- Host identifies interface and starts cable association process using 3 requests

SET_ASSOCIATION_INFORMATION	0x01
SET_ASSOCIATION_REQUEST	0x02
SET_ASSOCIATION_RESPONSE	0x03

- Everything is done over the USB cable, the UWB radio is not used



Step 2: Get Association Types

- Host gets list of wireless protocols that device supports (and wants to associate)
 - GET_ASSOCIATION_INFORMATION request
- Device returns list of the technologies it wants to associate
 - Certified Wireless USB is just one
- Two data structures
 - ASSOCIATION_INFORMATION
 - ASSOCIATION_REQUEST



Sample ASSOCIATION_INFORMATION

Offset	Field	Size	Value	Data
0	Length	2	Number	0019H
2	NumAssociationRequests	1	Number	02H
3	Flags	2	Bitmap	0000H
5	AssociationDataIndex	1	Number	01H
6	Reserved	1	Constant	00H
7	AssociationTypeId	2	Number	0001H
9	AssociationSubTypeId	2	Number	0000H
11	AssociationTypeInfoSize	4	Number	00000000H
15	AssociationDataIndex	1	Number	02H
16	Reserved	1	Constant	00H
17	AssociationTypeId	2	Number	0001H
19	AssociationSubTypeId	2	Number	0001H
21	AssociationTypeInfoSize	4	Number	00000033H

RetrieveHostInfo
(Step 3)

AssociateWUSB
(Step 4)

↑
Assumes FriendlyName "SAMPLE"
(Actual size varies 0x2c - 0x6c)



Step 3: RetrieveHostInfo

- Host sends its CHID to device
 - SET_ASSOCIATION_RESPONSE(RetrieveHostInfo)
 - May optionally send FriendlyName
- Host then asks device if it already has a CC for the host's CHID
 - GET_ASSOCIATION_REQUEST(AssociateWUSB)
 - Device returns DEVICE_INFO data structure
 - If device has a CC, it returns its CDID to the host
 - Else it returns zero
 - May optionally send FriendlyName & BandGroups
- Based on response, host can take appropriate user interface action (conditioning)

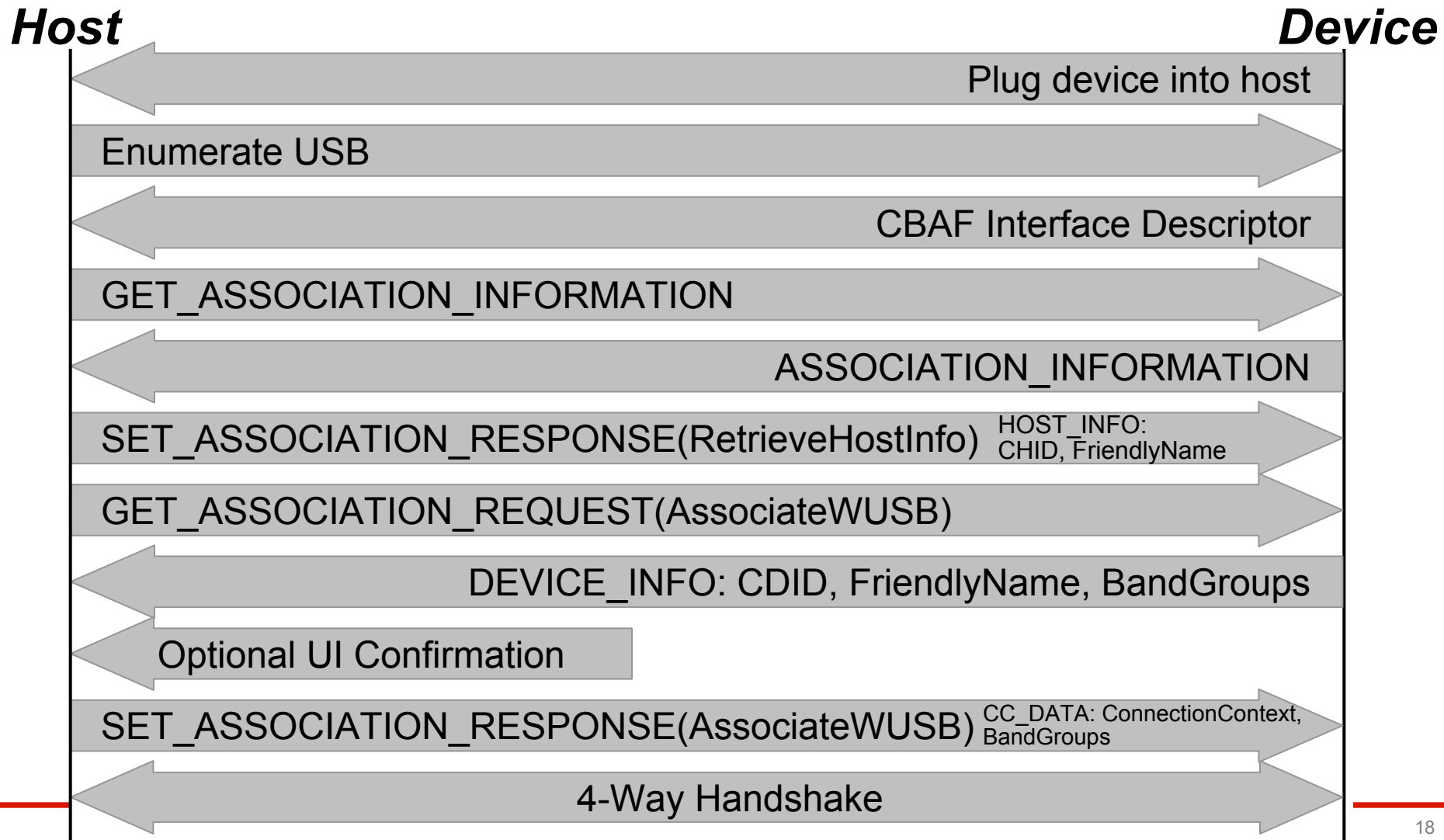


Step 4: AssociateWUSB

- Host creates CC and sends it to device
 - SET_ASSOCIATION_RESPONSE(AssociateWUSB)
 - CC_DATA data structure
- CC_DATA
 - Connection Context: CHID, CDID, CK
 - Band Groups supported by host
- Failure case has slightly different CC_DATA
 - AssociationStatus field specifies error reason

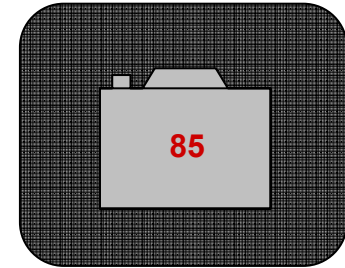
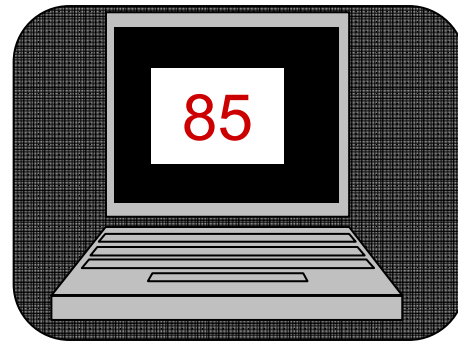


Cable Model Process - Summary





User Experience – Compare



You bring the device into the vicinity of the host.

From a menu on the host, you select the option to “add a new device.”

On the device, you select from a menu that says “connect to new host.”

The host screen displays a short code.

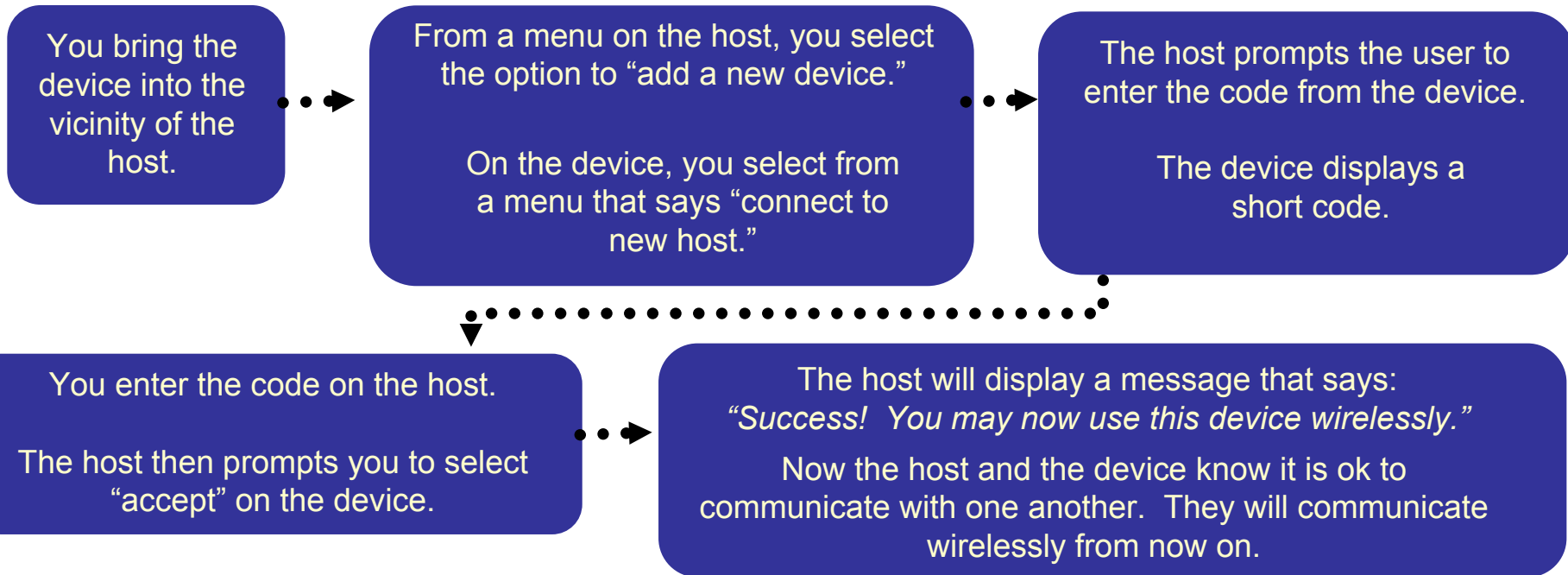
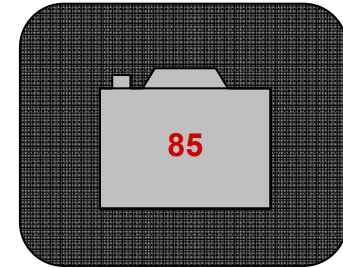
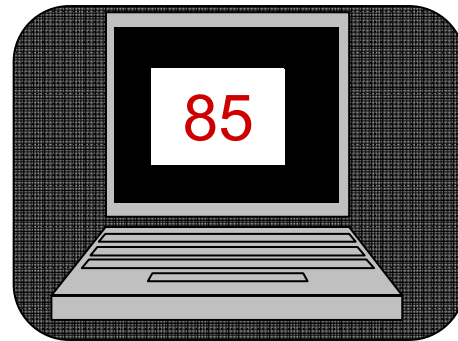
The device screen displays the same code.

You compare the codes and see they are the same, so you select “accept” on the host **and** then select “accept” on the device.

The host will display a message that says:
“Success! You may now use this device wirelessly.”
Now the host and the device know it is ok to communicate with one another. They will communicate wirelessly from now on.



User Experience – Enter



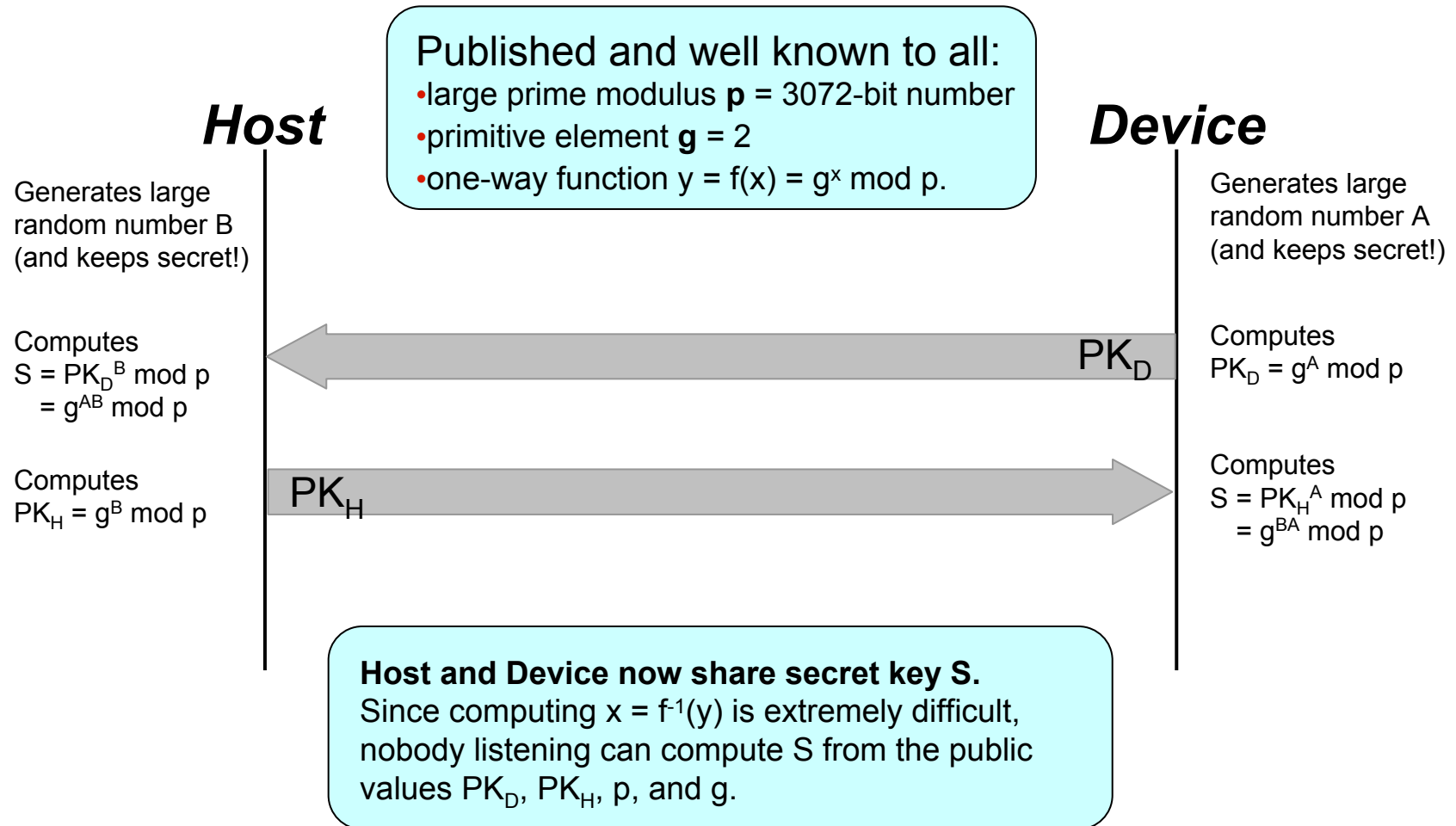


Security

- Unlike cable, numeric is not secure by nature
- Can be eavesdropped
 - Diffie-Hellman solves this
 - D-H requires gates or CPU power
- D-H doesn't protect against impersonators
 - Need to verify the identity of the host/device
 - Need a display on the device to do this

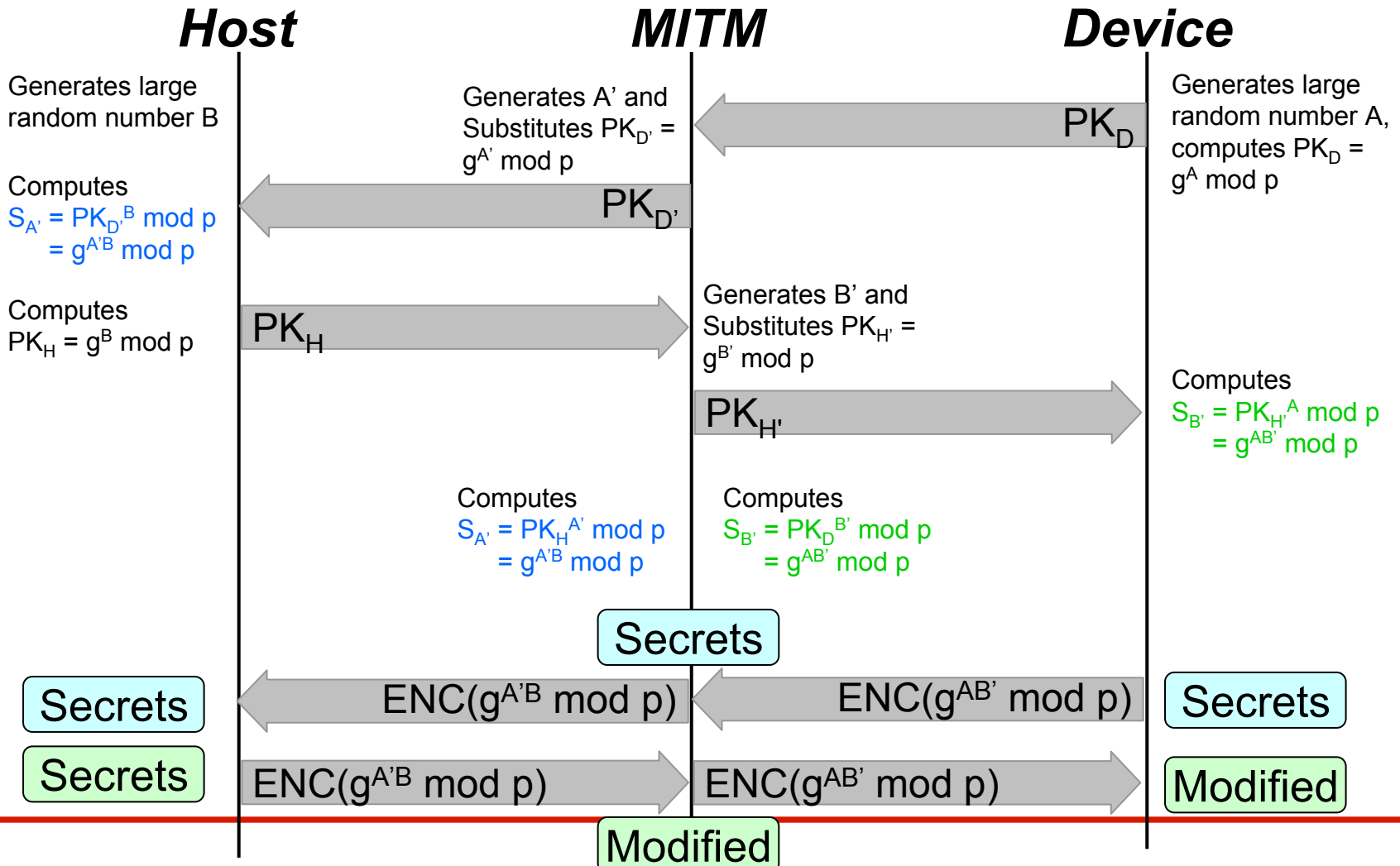


Diffie-Hellman Explained



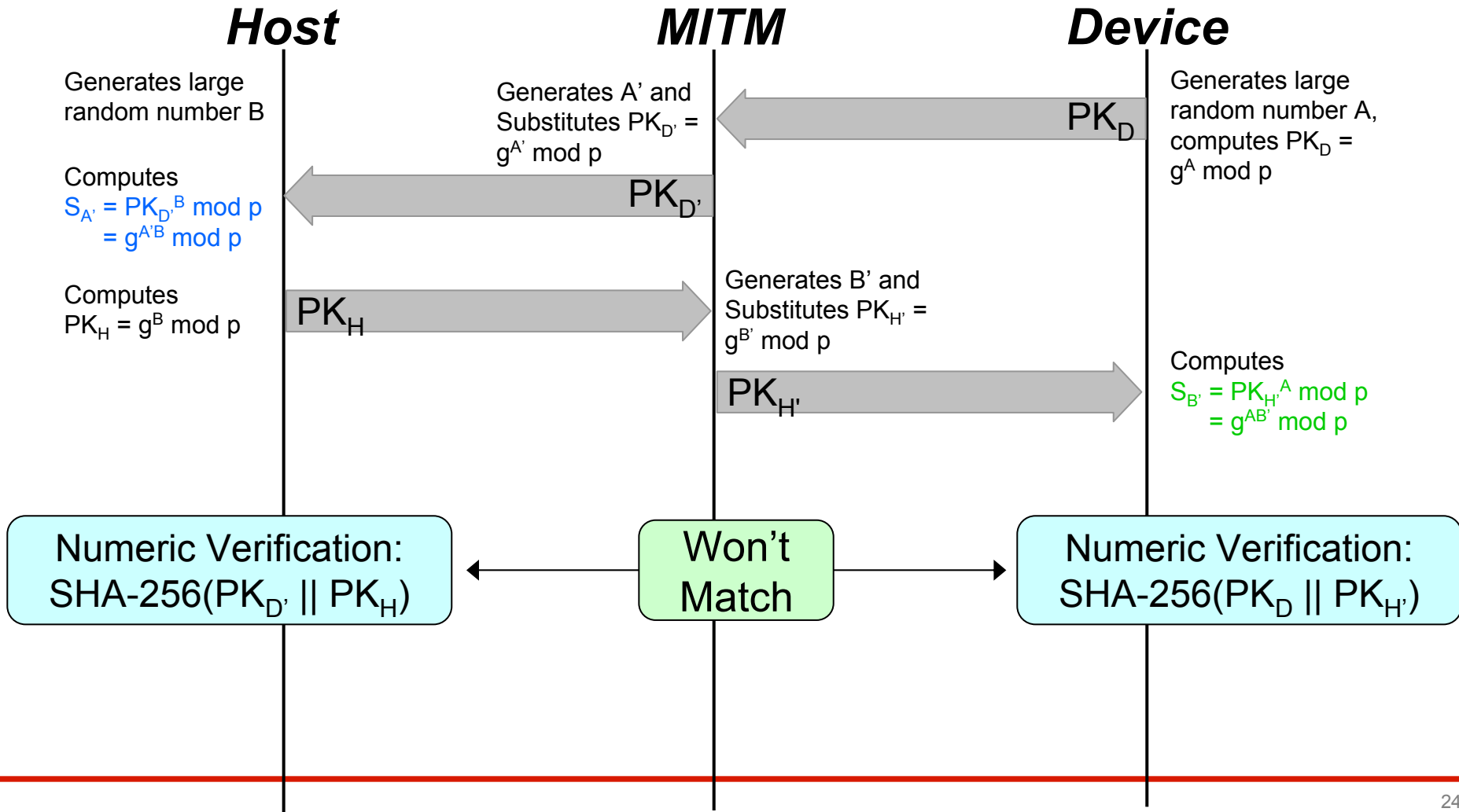


Man-in-the-Middle Problem





How Does Numeric Verify Help?





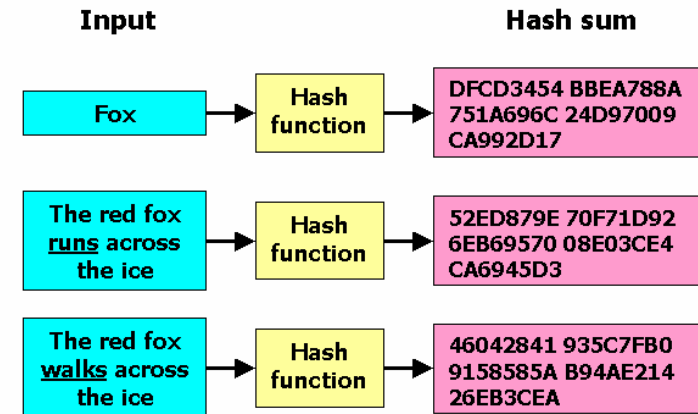
Other Crypto Fundamentals

- SHA-256

- Input: Message of any length
- Output: 256-bit message digest
- Random yet deterministic
- Difficult to reverse or find collisions

- HMAC-SHA-256

- Type of checksum or message authentication code
- Simultaneously verifies data integrity and authenticity of a message
- Input: Shared key and message to authenticate
- Output: 256-bit message digest
- Key and message protected against disclosure





A Word on Randomness

- Security **depends** on random numbers that are really random
- Must be derived from physical entropy source
- Must be freshly generated for each use
- $2 \leq \text{RandomNumber} \leq 2^L - 1$
 - L is number of random bits desired
- Must be unpredictable
 - Chosen with equal probability from available number space)



Example Use Cases

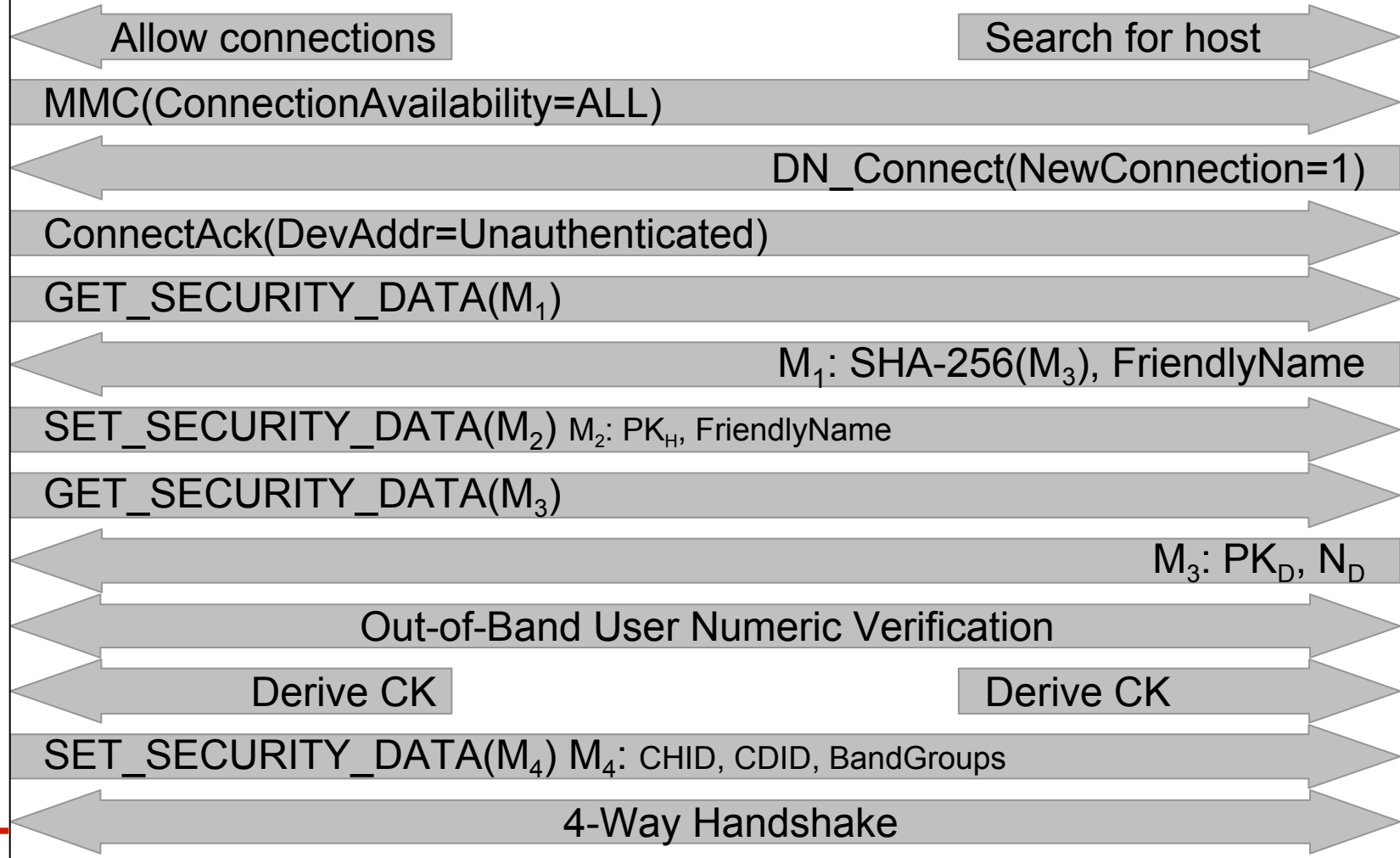
- Trivial scenarios – one owner, no attacker
 - 1 host and 1 device
 - 1 host and 2+ devices
 - 2+ hosts and 1+ devices
- Accidental scenarios
 - Any of the above scenarios + neighbor's host/device
 - User conditioning, verification, and low collision probability reduce chance of incorrect association
- Attack scenarios
 - Any of the above scenarios + active attacker
 - User verification prevents incorrect association
- User verification is the foundation of numeric model!



Numeric Model Process - Preview

Host

Device





Numeric Walkthrough

- Numeric model uses unencrypted, in-band requests
 - GET_SECURITY_INFO
 - SET_SECURITY_INFO
 - Host assigns temp DevAddr that only allows these two requests
 - IMPORTANT: Numeric model uses 4 messages (M_1 - M_4). These ARE NOT the same as the 4-way handshake!
- Process
 1. Starting the association
 2. Device hash commitment
 3. Public key exchange
 4. Compute shared secret
 5. Numeric verification
 6. Establish connection context



Step 1: Starting the Association

- Condition host to accept new devices
 - Host sets ConnectionAvailability=ALL in Host IE
- Condition device to look for new host
 - Device issues DN_Connect to host with NewConnection bit set
- Host responds ConnectAck and sets DevAddr to unauthenticated range (0x80+)
- Host and device are now able to communicate using GET_ and SET_ SECURITY_INFO



Step 2: Device Hash Commitment

- Host issues GET_SECURITY_DATA(1)
- Device generates random number A and calculates $PK_D = g^A \text{ mod } p$
 - Can be slow, can do earlier (power up)
- Device prepares M_3 data structure
 - M_3 : PK_D , N_D (number of digits to display)
- Device returns M_1 to host
 - M_1 : SHA-256(M_3), FriendlyName
- Hash commitment prevents attacks later



Step 3: Public Key Exchange

- Host sends its PK to device
 - Host generates random number B and calculates $PK_H = g^B \text{ mod } p$
 - Host issues SET_SECURITY_DATA(2)
 - Sends M_2 data structure (PK_H) to device
- Host gets device's PK
 - Host issues GET_SECURITY_DATA(3)
 - Device returns M_3 (from step 2): PK_D, N_D
 - Host computes SHA-256(M_3) and aborts if it doesn't match value from step 2 (prevents attacks)



Step 4: Compute Shared Secret

- Host and device both derive DHKey
 - Host: $\text{DHKey} = \text{SHA-256}(\text{PK}_D^B \text{ mod } p)$
 - Device: $\text{DHKey} = \text{SHA-256}(\text{PK}_H^A \text{ mod } p)$
- DHKey is shared secret only known to host and device
- Never sent over the air, completely secure
- DHKey computation can take a long time
- In parallel, host and device can continue with numeric verification

} Same!



Step 5: Numeric Verification

- Host and device both compute V
 - $V_D/V_H = \text{first 32 bits of SHA-256}(PK_D \parallel PK_H \parallel \text{“displayed digest”})$
- Device displays $V_D \bmod 10^N$ to user
- Host can either display or prompt
 - Display: Host shows $V_H \bmod 10^N$ to user
 - If values match, user presses “match” button
 - Else user presses “don’t match button”
 - Prompt: Host prompts user to enter device’s number
 - Host verifies that value entered matches $V_H \bmod 10^N$
- At the end of this process, host and device identities have been verified!



Step 6: Establish CC

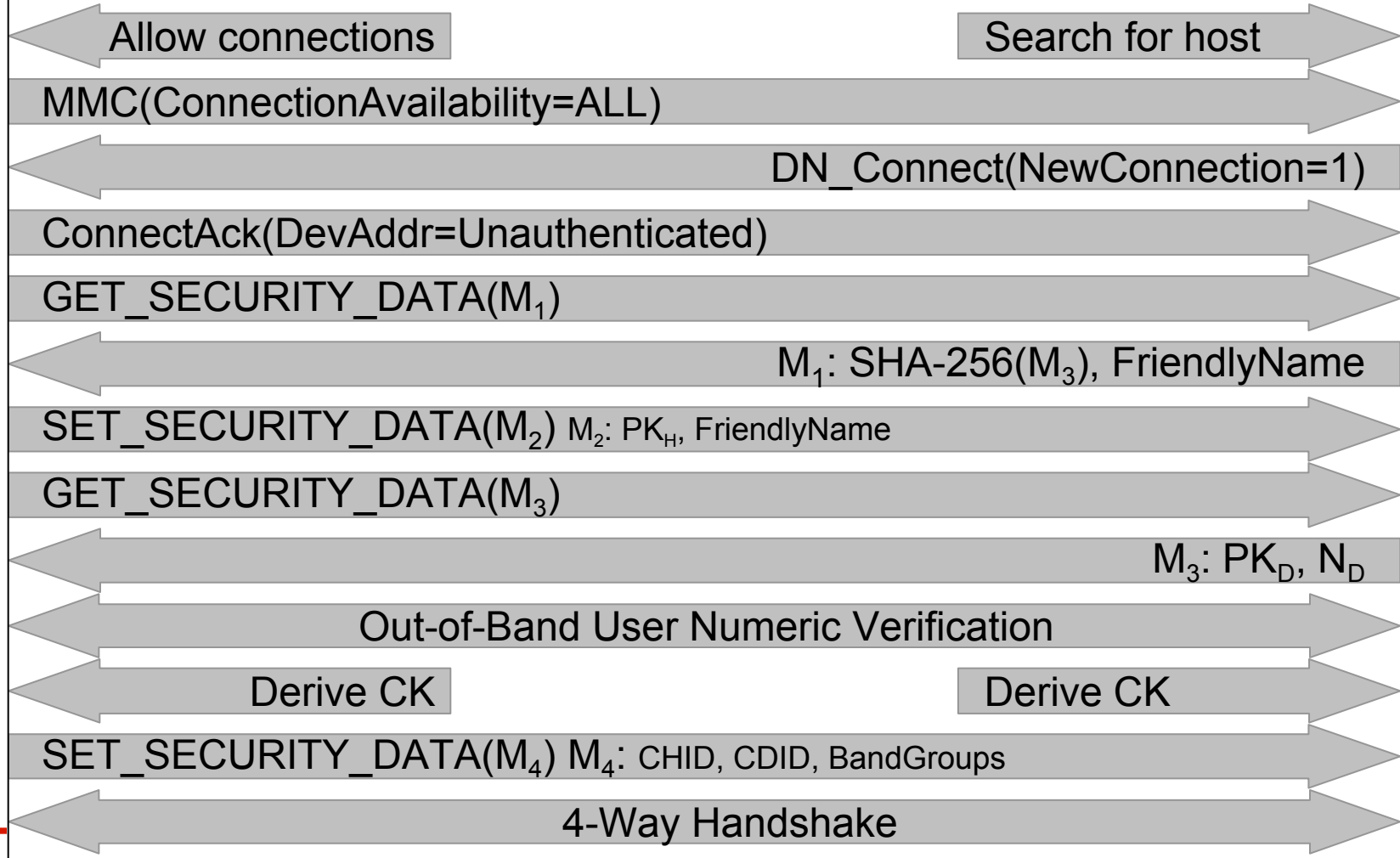
- Host and device compute $CK = \text{first 128 bits of HMAC-SHA-256}_{DHKey}$ (“connection key”)
 - Stored on each side independently
 - Never sent over the air
- Host sends M_4 containing other info to device
 - CHID, CDID, BandGroups
 - This info is not secret
- Device and host now proceed with 4-way handshake



Numeric Model Process - Review

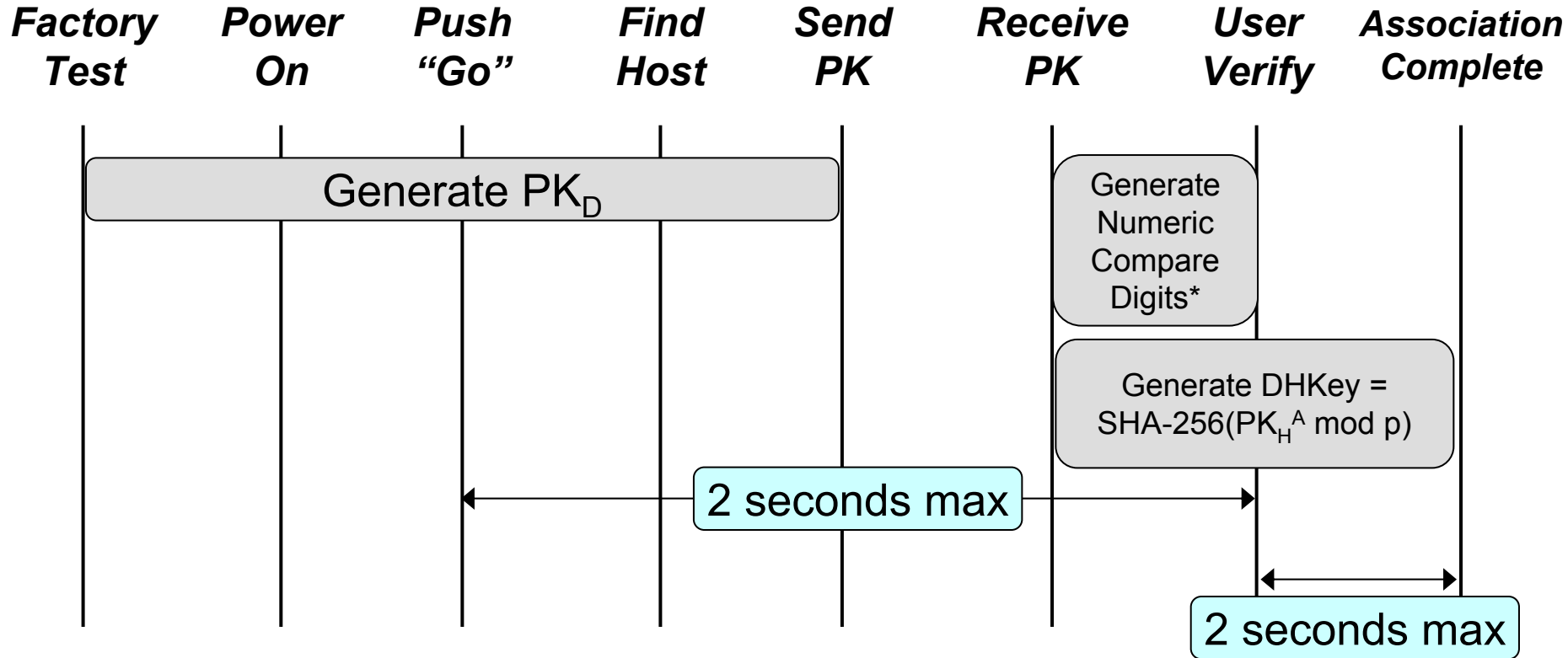
Host

Device





How Long Will It Take? (From Device Perspective)



* $SHA-256(PK_D \parallel PK_H \parallel \text{"displayed digest"})$

Implementation Gotchas



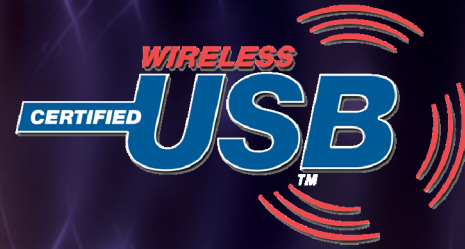
- Be careful!
- Many pitfalls in crypto
 - Random numbers not truly random or reused
 - Protocol errors
 - Poor garbage cleanup (temporary vars)
- Remember the user in your designs
- Follow the spec carefully
- Display size: Minimum 2 digit, more is better!
- Use the test vectors!

Summary



- Cable and numeric models
- Association supplement publicly available
 - www.usb.org/wusb
 - A few errata will be published next month
- Implement cable and numeric models in your products now!
- Check out the technology showcase demos

And a brief advertisement...



Association FAQ

- How is disassociation defined?
- What happens if a device discovers a valid host in range, but its CC for that host doesn't work any more?
- Are vendor-specific association models allowed (in addition to the mandatory models)?
- What are the rules and regulations regarding the export of products containing strong cryptography?
- How many CCs should a device store?
- And many more... see Association FAQ at www.usb.org/wusb (part of core spec ZIP file download)

Thank you!



Developers Conference 2006

Taipei, Taiwan

Backup





Association Timeline

11/2002	Association marketing requirements
09/2003	End-user focus groups Association working group meetings Developer F2F meetings
11/2004	Industry Update (SF)
01/2005	Industry Update (Milpitas)
04/2005	Second focus group study
05/2005	San Jose DevCon
09/2005	Version 0.9 / Tokyo DevCon
12/2005	Final review
03/2006	Association Model Supplement 1.0 released publicly
Q3/2006	Test specification available
2007+	Alignment with other wireless specs, new technologies (i.e., NFC)

Wired vs. Wireless Association

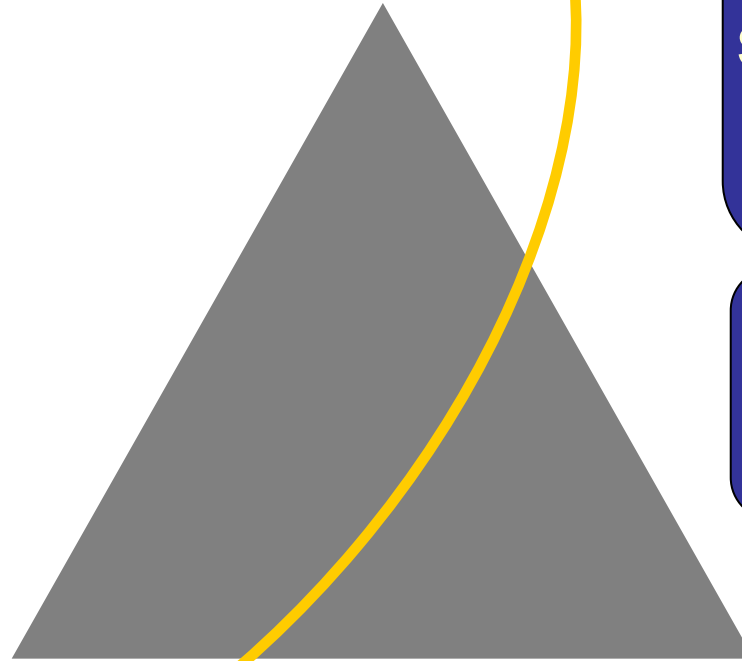


Issue	Wired	Wireless
Accidental connection	Impossible	Easy
Conditioning	Easy	Can be difficult
Passive attack	Very hard	Easy
Active attack (MITM)	Impossible	Unknown
User intent	Obvious	Can be difficult

Association Models Supplement addresses these issues for Certified Wireless USB

Tough Decision

Strong Security



Ease-of-Use

Must not
sacrifice security
or usability!

(But obviously can't
cost **too** much.)

No Extra Cost

Diffie-Hellman Software Implementation



- Example Implementation, C source code
- 200MHz ARM (XScale-PXA255)
- $g^A \bmod p \rightarrow 0.246$ seconds
- $PK_H^A \bmod p \rightarrow 0.327$ seconds
- Code size: 4076 bytes (thumb mode, -O3)
- Using Phil Zimmermann's bnlb

Cabled Devices



Products with upstream USB Ports

- **Should** behave as a regular wired USB product while plugged in
 - **Should** enumerate at least 1 class in addition to association class
- **Must** support cable association
 - **Must** enumerate as association class
- **Should not** enumerate with the same host wirelessly while connected via a wire

Focus Group Study



- Conducted May 2005 in 2 markets
- 5 groups, 6 participants each
- Professionally moderated
- Goal was to assess perceived usability, security of 5 models

Focus Group Results



Wireless Expectations – Emphasis on Ease of Use

- Should be simple & easy
- Consumers hope/expect security will be there

Connection Scenarios – Most Popular: NFC and Numeric

- NFC has the biggest “wow” factor
 - “This is the ease I’m looking for!”
 - Some worried that it was insecure because it was too easy
- Numeric Model seen as all-around strong performer
 - Easy, reassuring, secure.
- Cable Model neutral
 - Redeemed itself in the security category
 - Less sophisticated users thought it was just fine

Focus Group Results



Purchase Motivator or Deterrent?

	Numeric	PIN	NFC	Cable	LED
Motivator 😊	13	14	17	10	1
Neutral 😐	16	15	8	12	8
Deterrent ☹️	1	1	5	8	21

Most/Least Favorite

	Numeric	PIN	NFC	Cable	LED
Ranked #1 😊	10	5	13	0	0
Ranked #2-4 😐	20	22	14	24	16
Ranked #5 ☹️	0	3	3	6	14

Introducer Device



- Used for non-mobile host/device scenarios
- First plug introducer into host
 - Host enumerates CBAF on introducer
 - Stores CC on introducer
- Then plug into device
 - Introducer enumerates CBAF on device
 - Stores CC on device
- Mass storage/flash drive option may also be viable



Developers Conference 2006

Taipei, Taiwan