

Improving Wireless Privacy with an Identifier-Free Link Layer Protocol

Ben Greenstein, Damon McCoy, Jeffrey Pang,
Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall

Intel Research Seattle, University of Colorado,
Carnegie Mellon University, University of Washington

Our Wireless World



Link Layer Header

Blood pressure: high

Link Layer Header

PrivateVideo1.avi

Link Layer Header

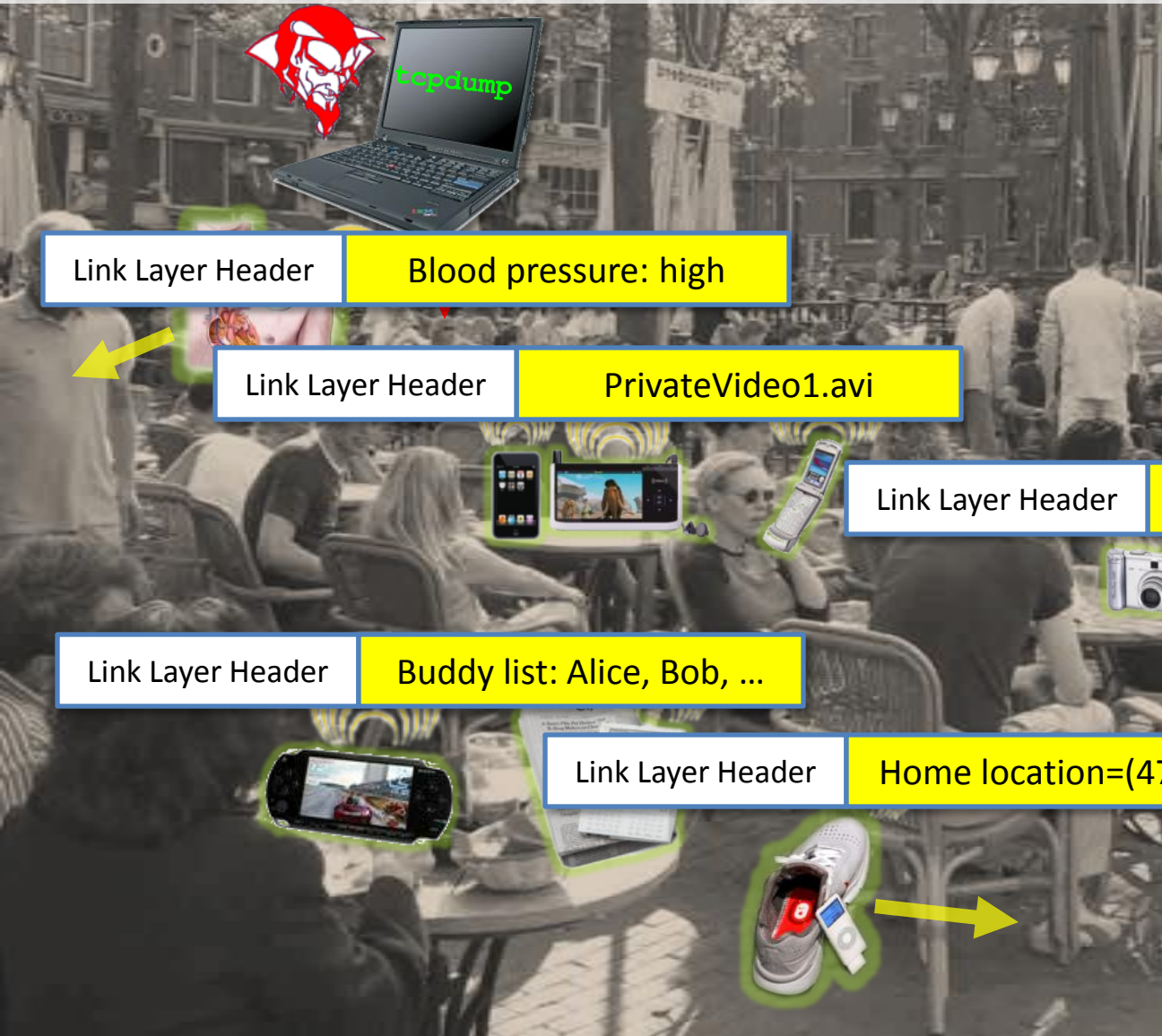
PrivatePhoto1.jpg

Link Layer Header

Buddy list: Alice, Bob, ...

Link Layer Header

Home location=(47.28,...



Best Security Practices

Bootstrap

Username: Alice
Key: 0x348190...

SSID: Bob's Network
Key: 0x2384949...

Out-of-band (e.g., password, WiFi Protected Setup)

Discover

802.11 probe

Is Bob's Network here?

802.11 beacon

Bob's Network is here

Authenticate and Bind

802.11 auth

Proof that I'm Alice

802.11 auth

Proof that I'm Bob

Send Data

802.11 header

802.11 header

- Confidentiality
- Authenticity
- Integrity



Privacy Problems Remain

Many exposed bits are (or can be used as) identifiers that are linked over time



Discover

802.11 probe

Is Bob's Network here?

802.11 beacon

Bob's Network is here

Authenticate
and Bind

802.11 auth

Proof that I'm Alice

802.11 auth

Proof that I'm Bob

Send Data

MAC addr, seqno, ...

MAC addr, seqno, ...

- Confidentiality
- Authenticity
- Integrity



Problem: Long-Term Linking

802.11 beacon

Alice's iPod is here

MAC: 12:34:56:78:90:ab



802.11 beacon

Alice's iPod is here

MAC: 12:34:56:78:90:ab



802.11 probe

Is Alice's iPod here?



Easy to identify and relate devices over time

Problem: Long-Term Linking

Linking enables location tracking, user profiling, inventorying, relationship profiling, ...

[Greenstein, *HotOS '07*; Jiang, *MobiSys '07*; Pang, *MobiCom '07*, *HotNets '07*]



cnet NEWS.com

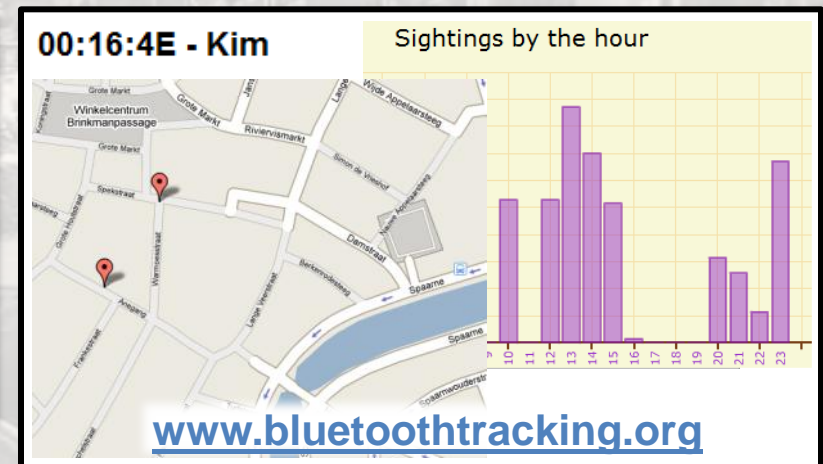
Search:

Today on CNET | Reviews | News | Downloads | Tips & Tricks | CNET TV | Compare Prices | Blogs

Business Tech | Cutting Edge | Access | Threats | Media 2.0 | Markets | Personal Tech | News Blogs | Video | Extra

Wireless location tracking draws privacy questions

Wireless products that can do everything from tracking your children to finding you a nearby date this weekend seem to fall outside the scope of federal privacy laws, and that may need to change, an industry group said.



802.11 header

Is "djw" here?

"djw" is here

47.679741, -122.295578

Home

www.wigle.net

Phone pirates in seek and steal mission

MOBILE phone technology is being used by thieves to seek out and steal laptops locked in cars in Cambridgeshire.

Forcetracker
Stolen Vehicle Tracking
No Monthly Fee
www.forcetracker.org

Up-to-date mobiles often have Bluetooth technology which allows other compatible devices, including laptops, to link up and exchange information, and log

Problem: Short-Term Linking

3-9 data streams overlap each 100 ms, on average (see paper)

12:34:56:78:90:ab, seqno: 1, ...

12:34:56:78:90:ab, seqno: 2, ...

00:00:99:99:11:11, seqno: 102, ...

12:34:56:78:90:ab, seqno: 3, ...

00:00:99:99:11:11, seqno: 103, ...

12:34:56:78:90:ab, seqno: 4, ...

00:00:99:99:11:11, seqno: 104, ...

Easy to isolate distinct packet streams

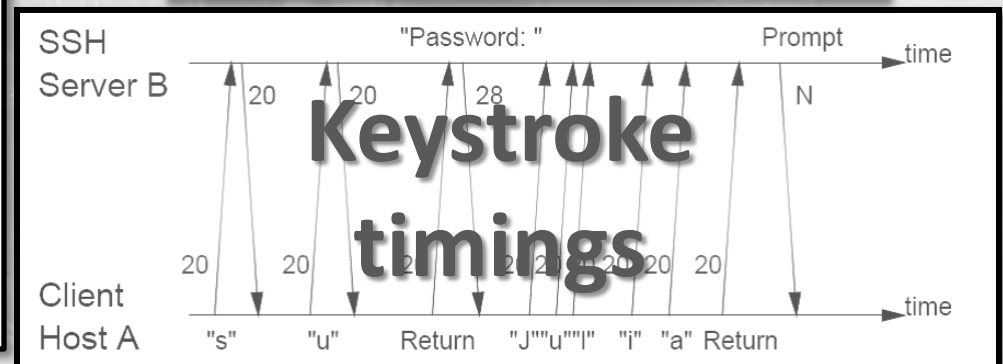
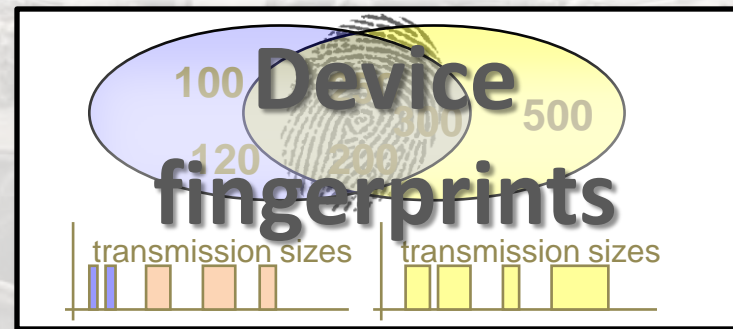
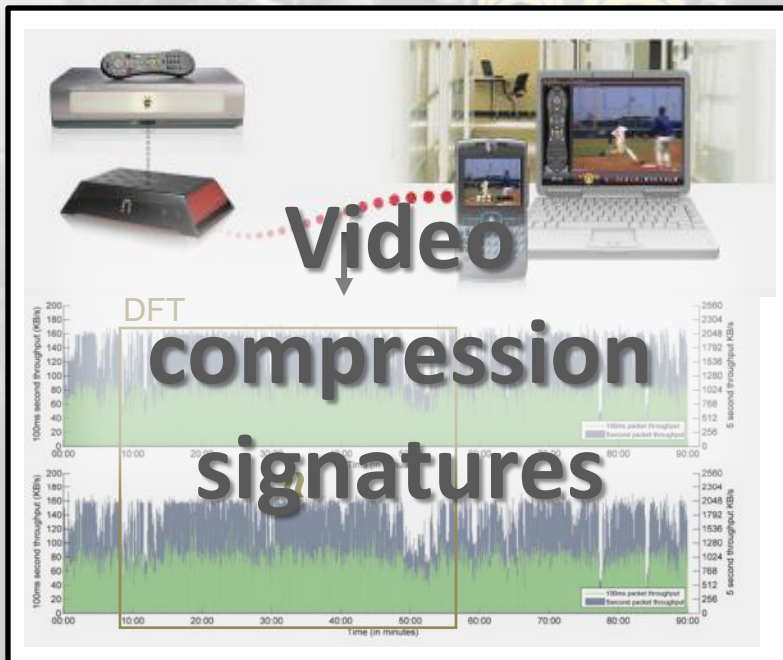


Problem: Short-Term Linking

Isolated data streams are more susceptible to side-channel analysis on packet sizes and timing

- Exposes keystrokes, VoIP calls, webpages, movies, ...

[Liberatore, CCS '06; Pang, MobiCom '07; Saponas, Usenix Security '07; Song, Usenix Security '01; Wright, IEEE S&P '08; Wright, Usenix Security '07]



Fundamental Problem

Many exposed bits are (or can be used as) identifiers that are linked over time

Discover

802.11 probe

Is Bob's Network here?

802.11 beacon

Bob's Network is here

Authenticate
and Bind

802.11 auth

Proof that I'm Alice

802.11 auth

Proof that I'm Bob

Send Data

MAC addr, seqno, ...

MAC addr, seqno, ...



Goal: Make All Bits Appear Random

Bootstrap

SSID: Bob's Network
Key: 0x2384949...

Username: Alice
Key: 0x348190...

Discover

Authenticate
and Bind

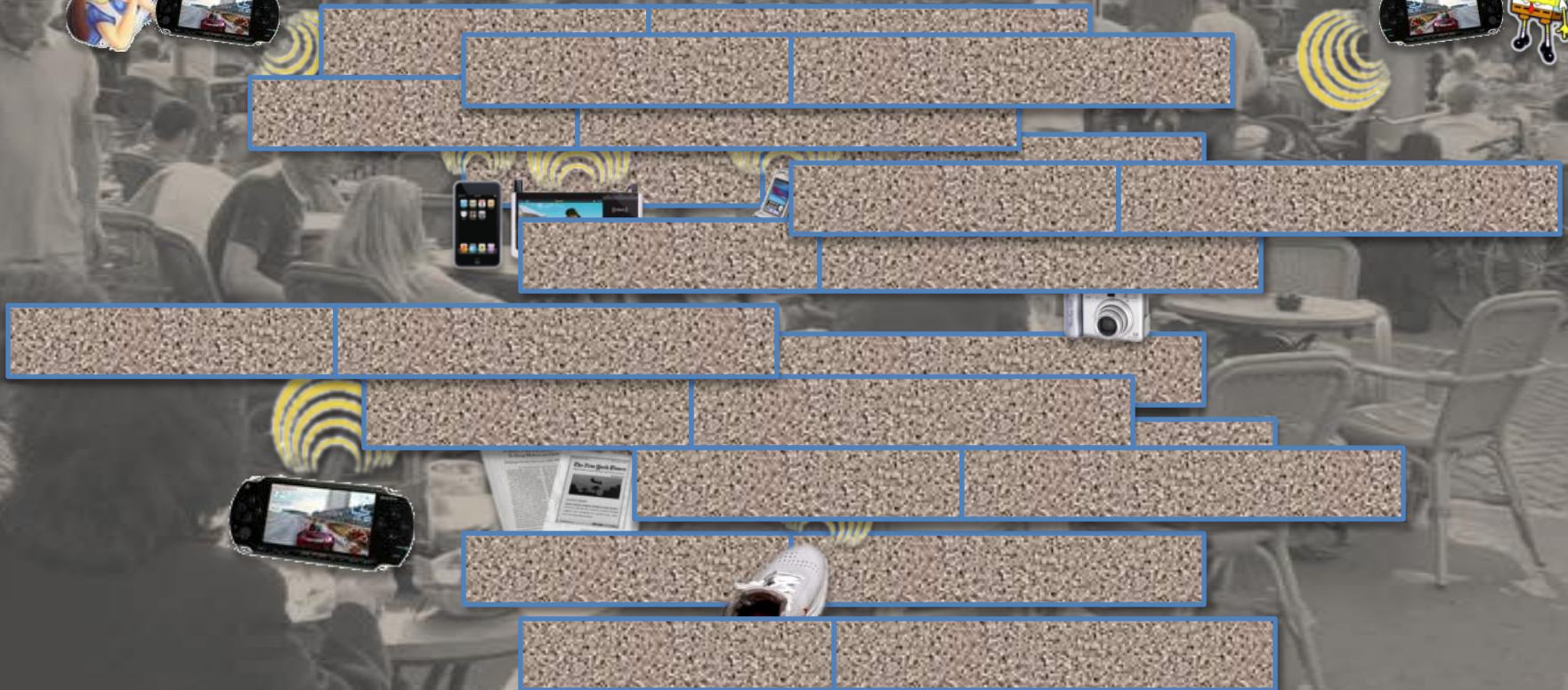
Send Data



Challenge: Filtering without Identifiers

Which packets are mine?

Which packets are mine?



Talk Overview

- Motivation and Goals
- Design Requirements
- Straw man: MAC Pseudonyms
- Straw man: Encrypt Everything
- Solution: SlyFi

Goal: This Protocol

Bootstrap

SSID: Bob's Network
Key: 0x2384949...

Username: Alice
Key: 0x348190...



Discover



Authenticate
and Bind




Send Data



Design Requirements



- When *A* generates *Message* to *B*, she sends:

$$\text{PrivateMessage} = F(A, B, \text{Message})$$


where **F** has these properties:

- **Confidentiality:** Only *A* and *B* can determine *Message*.
- **Authenticity:** *B* can verify *A* created *PrivateMessage*.
- **Integrity:** *B* can verify *Message* not modified.
- **Unlinkability:** Only *A* and *B* can link *PrivateMessages* to same sender or receiver.
- **Efficiency:** *B* can process *PrivateMessages* as fast as he can receive them.







Solution Summary

	Confidentiality	Authenticity	Integrity	Unlinkability	Efficiency
802.11 WPA	Only Data Payload	Only Data Payload	Only Data Payload		
MAC Pseudonyms					
Public Key Symmetric Key					
SlyFi: Discovery/Binding					
SlyFi: Data packets					

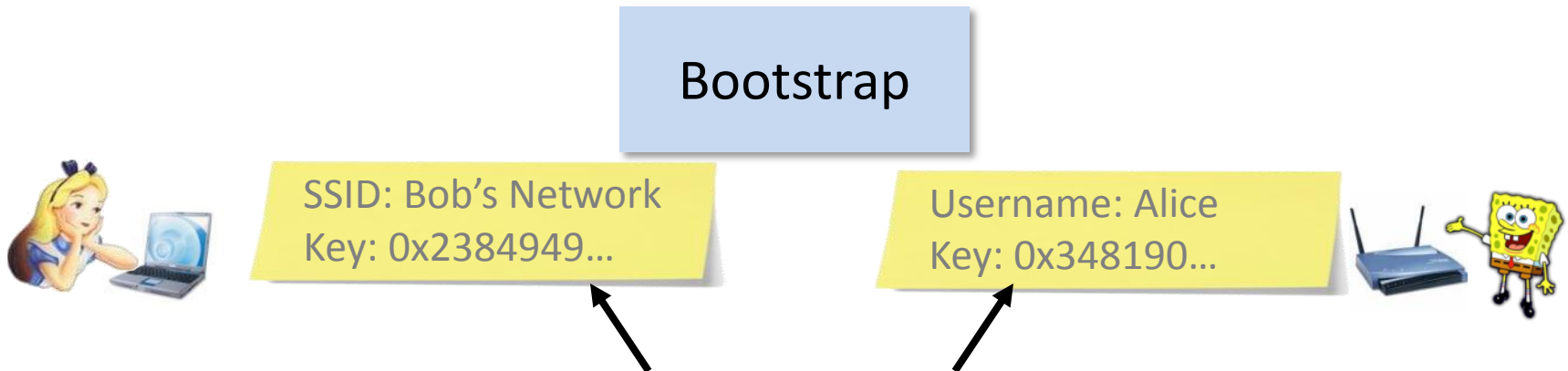
Straw man: MAC Pseudonyms

- **Idea:** change MAC address periodically
 - Per session or when idle [Gruteser '05, Jiang '07]
- **Other fields remain (e.g., in discovery/binding)**
 - No mechanism for data authentication/encryption
 - Doesn't hide network names during discovery or credentials during authentication
- **Pseudonyms are linkable in the short-term**
 - Same MAC must be used for each association
 - Data streams still vulnerable to side-channel leaks

Solution Summary

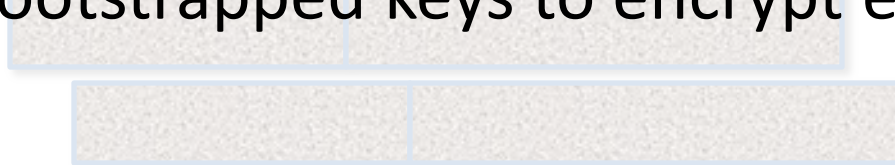
	Confidentiality	Authenticity	Integrity	Unlinkability	Efficiency
802.11 WPA	Only Data Payload	Only Data Payload	Only Data Payload		
MAC Pseudonyms				Long Term	
Public Key					
Symmetric Key					
SlyFi: Discovery/Binding					
SlyFi: Data packets					

Straw man: Encrypt Everything

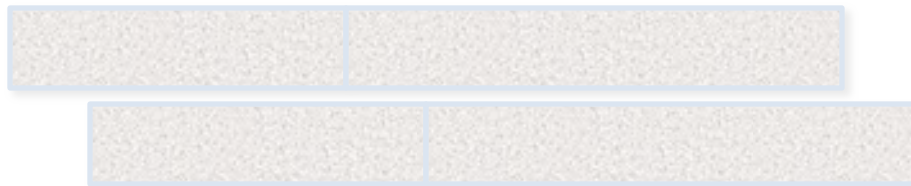


Idea: Use bootstrapped keys to encrypt everything

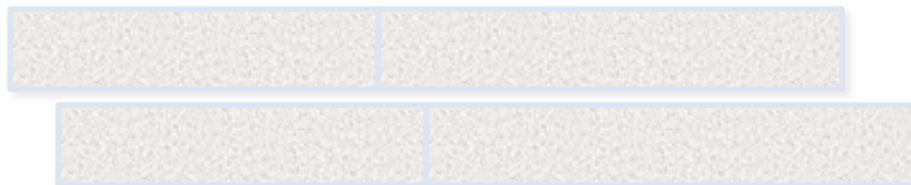
Discover



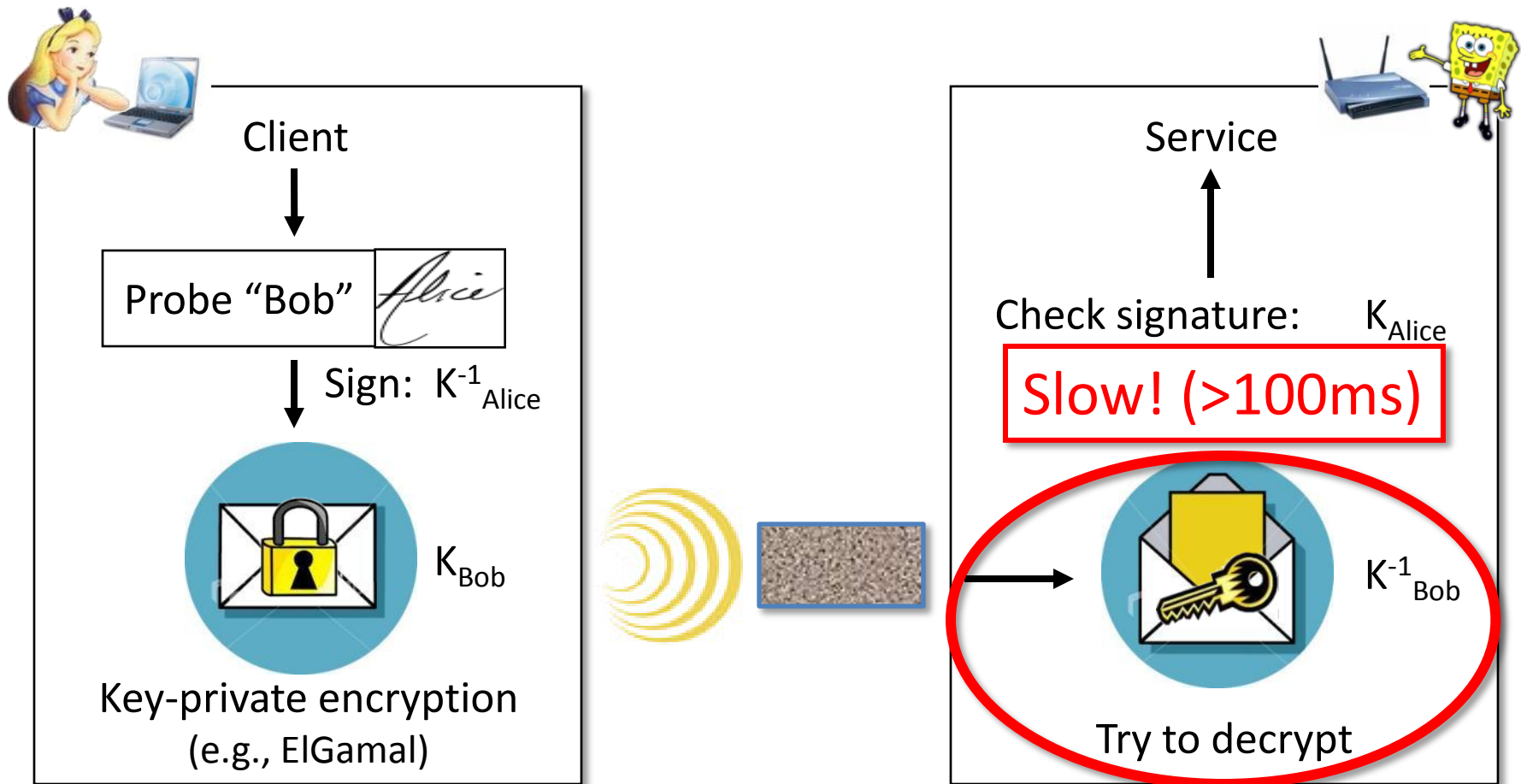
Authenticate
and Bind



Send Data

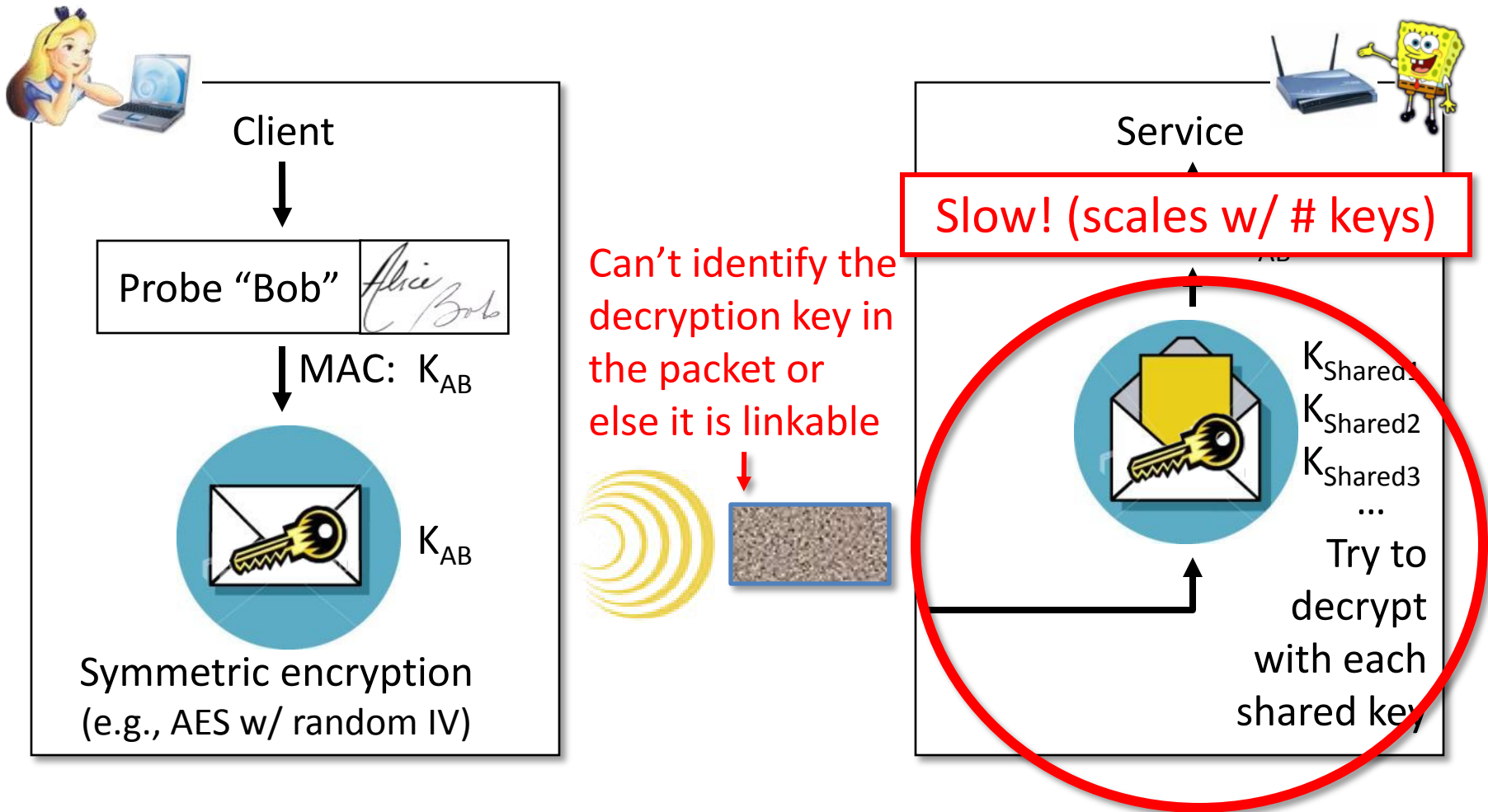


Straw man: Public Key Protocol














Based on [Abadi '04]

Straw man: Symmetric Key Protocol



Different symmetric key per potential sender

Solution Summary

	Confidentiality	Authenticity	Integrity	Unlinkability	Efficiency
802.11 WPA	Only Data Payload	Only Data Payload	Only Data Payload		
MAC Pseudonyms				Long Term	
Public Key Protocol					
Symmetric Key Protocol					
SlyFi: Discovery/Binding					
SlyFi: Data packets					

SlyFi

- Symmetric key almost works, but tension between:
 - Unlinkability: can't expose the identity of the key
 - Efficiency: need to identify the key to avoid trying all keys
- **Idea:** Identify the key in an unlinkable way
- Approach:
 - Sender **A** and receiver **B** agree on tokens: $T_1^{AB}, T_2^{AB}, T_3^{AB}, \dots$
 - **A** attaches T_i^{AB} to encrypted packet for **B**

SlyFi

Required properties:

- Third parties can not link T_i^{AB} and T_j^{AB} if $i \neq j$
- **A** doesn't reuse T_i^{AB}
- **A** and **B** can compute T_i^{AB} independently

Main challenge:

Sender and receiver must synchronize i

Symmetric encryption
(e.g., AES w/ random IV)

$$T_i^{AB} = \text{AES}_{K_{AB}}(i)$$

Lookup T_i^{AB} in a
table to get K_{AB}

$$T_i^{AB} = \text{AES}_{K_{AB}}(i)$$

SlyFi: Data Transport

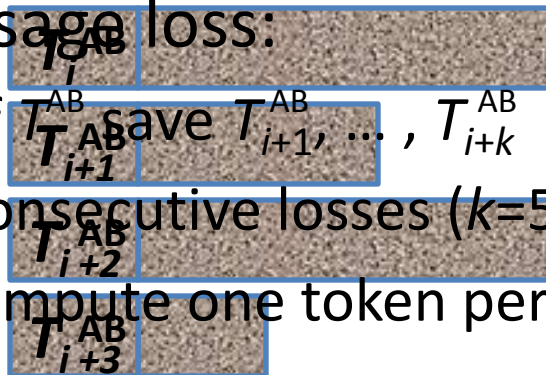
- Data messages:
 - Only sent over established connections
 - ⇒ Expect messages to be delivered
 - ⇒ Use implicit transmission number to synchronize i

$$T_i^{AB} = \text{AES}_{K_{AB}}(i) \quad \text{where } i = \text{transmission \#}$$

- On receipt of T_i^{AB} , **B** computes next expected: T_{i+1}^{AB}

- Handling message loss:

- On receipt of T_i^{AB} save $T_{i+1}^{AB}, \dots, T_{i+k}^{AB}$ in table
- Tolerates k consecutive losses ($k=50$ is enough [Reis, 06])
- No loss ⇒ compute one token per reception



SlyFi: Discovery/Binding

- Discovery & binding messages:
 - Often sent when other party is not present
 - ⇒ Can't expect most messages to be delivered
 - ⇒ Can't rely on transmission reception to synchronize i



T_i^{AB} Is Bob's Network here?

:

T_{i+1}^{AB} Is Bob's Network here?

:

T_{i+2}^{AB} Is Bob's Network here?

:

T_{i+3}^{AB} Is Bob's Network here?

Nope.

Nope.

Nope.

$i = ?$



SlyFi: Discovery/Binding

- Discovery & binding messages:
 - **Infrequent**: only sent when trying to associate
 - **Narrow interface**: single application, few side-channels
- ⇒ Linkability at short timescales is usually OK
- ⇒ Use loosely synchronized time to synchronize i

$$T_i^{AB} = \text{AES}_{K_{AB}}(i) \quad \text{where } i = \lfloor \text{current time} / 5 \text{ min} \rfloor$$

- At the start of time interval i , compute T_i^{AB}

- Handling clock skew
 -      

- Tolerates clock skew

SlyFi: Other Protocol Details

- Broadcast
- Higher-layer binding
- Time synchronization
- Roaming
- Coexistence with 802.11
- Link-layer ACKs
- Preventing replay attacks
- etc.



See paper

Performance Evaluation

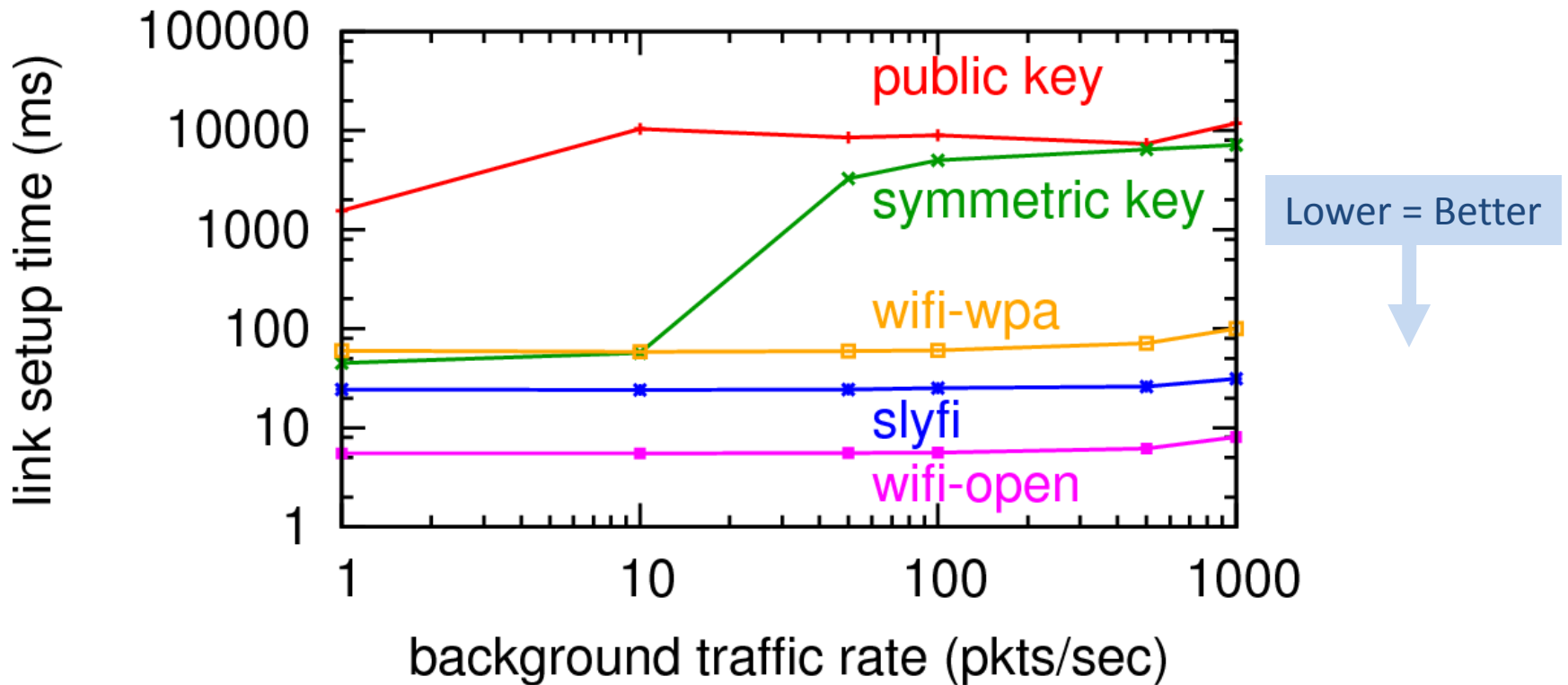
- SlyFi implementation:
 - Linux kernel module using Click Modular Router
 - Run on Soekris devices (similar to APs, iPods, etc.)
- Comparison protocols:
 - **wifi-open:** 802.11 with no security
 - **wifi-wpa:** 802.11 with WPA PSK/CCMP
 - **public-key:** straw man
 - **symmetric-key:** straw man
 - **armknecht:** previous header encryption proposal

Similar {

Experiment:

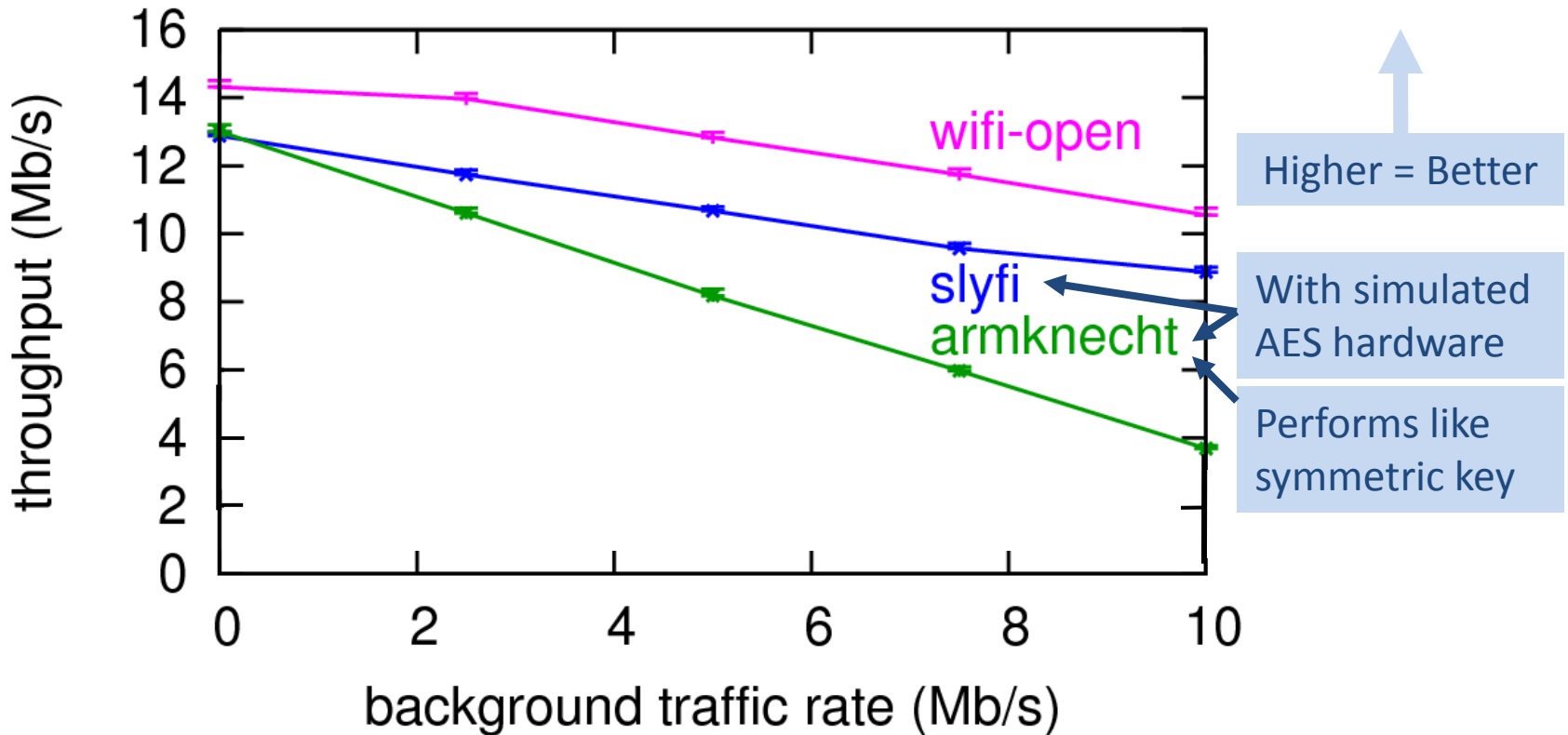


Discovery/Binding Time























SlyFi link setup has less overhead than WPA

Data Throughput



SlyFi data filtering is about as efficient as 802.11

Solution Summary

	Confidentiality	Authenticity	Integrity	Unlinkability	Efficiency
802.11 WPA	Only Data Payload	Only Data Payload	Only Data Payload		
MAC Pseudonyms				Long Term	
Public Key Symmetric Key					
SlyFi : Discovery/Binding				Long Term	
SlyFi : Data packets					

Conclusion

- Wireless devices are becoming personal and pervasive
- Best practices don't protect users from simple attacks
 - Long-term linking: tracking, profiling, inventorying
 - Short-term linking: side-channel attacks
- SlyFi makes these attacks much more difficult to do
 - Removes all bits that are (or can be used as) identifiers







<http://tw.seattle.intel-research.net>

===== CONTEXT =====

Related Work

- Private discovery
 - Public key straw man [Abadi, '04]
 - Private discovery sketch [Pang '07]
 - Privately announce existence to friends [Cox '07]
 - Different application; uses hash chain
- Encrypted data transport headers
 - Must try all keys to filter [Armknrecht '07]
 - Targeted at WPANs and use hash chains [Singelee '06]
- SlyFi is the first complete protocol and implementation

802.11w: Protected Management Frames

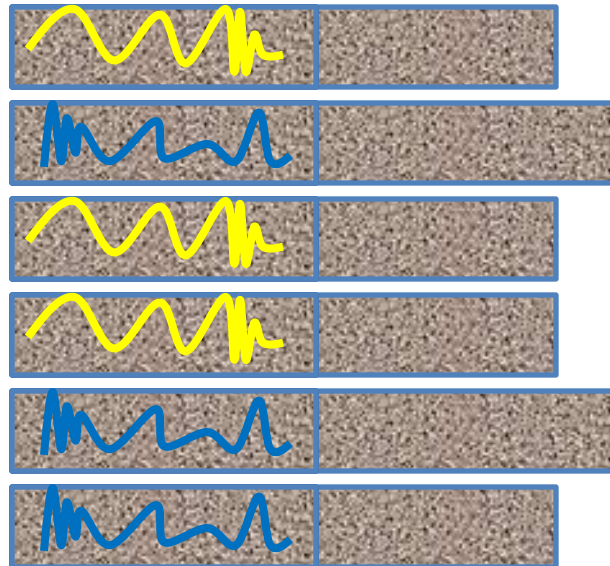
	Confidentiality	Authenticity	Integrity	Unlinkability	Efficiency
802.11i (WPA)	Data Payload	Data Payload	Data Payload		✓
802.11i + 802.11w	Unicast Frames	✓	✓		✓
MAC Pseudonyms				Long Term	✓
Public Key Symmetric Key	✓	✓	✓	✓	
SlyFi : Discovery/Binding	✓	✓	✓	Long Term	✓
SlyFi : Data packets	✓	✓	✓	✓	✓

Why not GSM Pseudonyms?

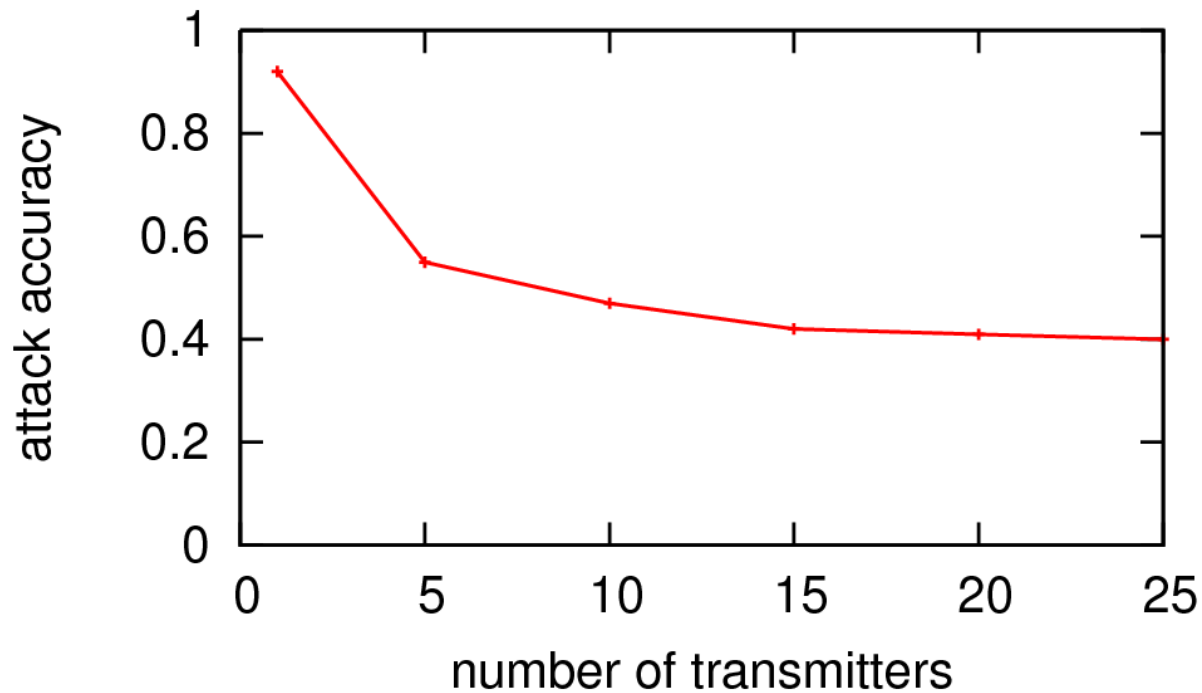
- GSM pseudonym properties
 - Provider must assign new pseudonym to client to change it
 - Only a single application used on GSM network
- GSM pseudonyms not sufficient when
 - Both parties in discovery want to be private
 - May require using pseudonym when the provider is not present (e.g., during discovery)
 - Many applications with many side-channels
 - Must accommodate device heterogeneity, evolution

PHY Layer and Timing Signatures

- PHY layer and timing signatures remain
- These are not as accurate and can require uncommon or expensive hardware
- Obscuring these signatures is future work
- SlyFi raises the bar and is a necessary first step



Linking with Signal Strength

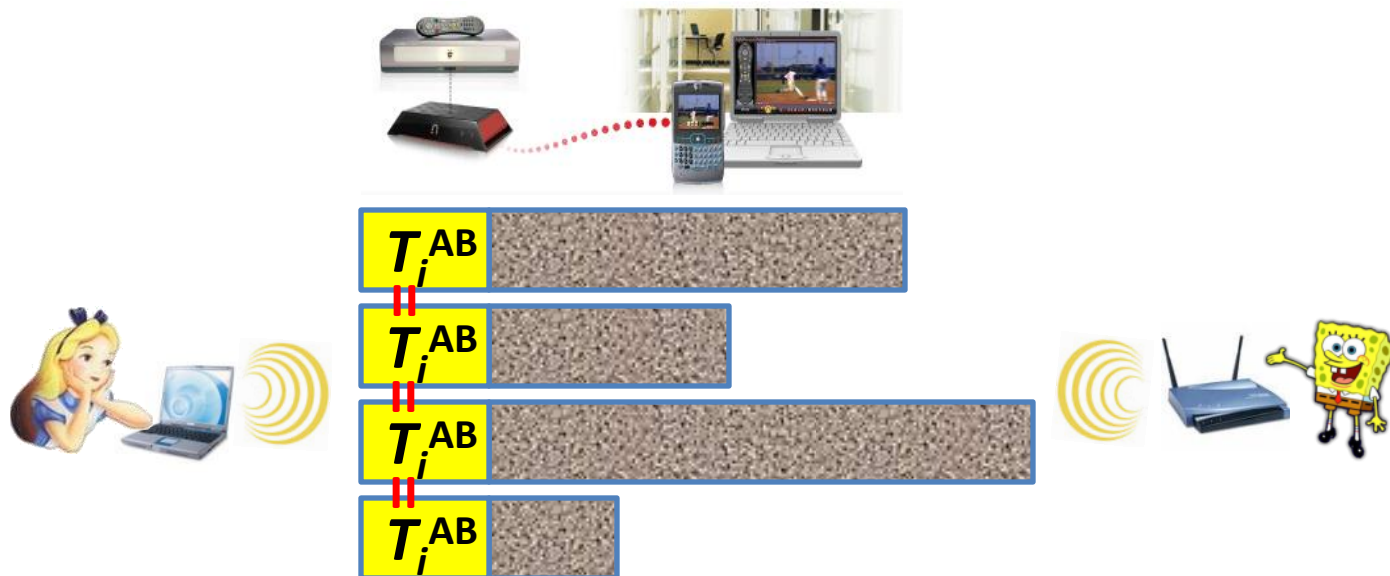


- **Attack:** website finger-printing using [Liberatore CCS '06]
- Attacker has 5 nodes to record packets' RSSIs
- Attacker uses k -means clustering to determine which packets belong to each client. Set of RSSIs is the feature vector.
- Experiment conservatively assumes that attacker knows k
- Clustering accuracy $> 75\%$ for all experiments

Side-channel attack accuracy degrades significantly even if attacker tries to use signal strength to link packets

Why not Time for Data Transport?

- Data messages:
 - **Frequent**: sent often to deliver data
 - **Wide interface**: many applications, many side-channels
- ⇒ Linkability at short timescales is **NOT** usually OK
- ⇒ Can **NOT** use loosely synchronized time to synchronize i



Future Work

- Private, automated bootstrapping [Greenstein, Pang]
 - Leverage transitive trust relationships
 - Leverage device reputation, measurable context
- Measuring/defending against PHY layer linking [McCoy]
 - Leverage transmit power control, directional antennas
- Masking remaining timing side-channels [Pang]
 - Perform intelligent packet padding/cover traffic

Packet Format

Token

Unencrypted Message

$s = \{addr_{AB}^i, AES_{k_{AB}^{Enc}}(k_p)\}$	$mac = AES-CMAC_{k_{AB}^{MAC}}(s)$	$etext = AES-CBC_{k_{p1}}(p)$	$emac = AES-CMAC_{k_{p2}}(etext)$
32 bytes	16 bytes	variable	16 bytes

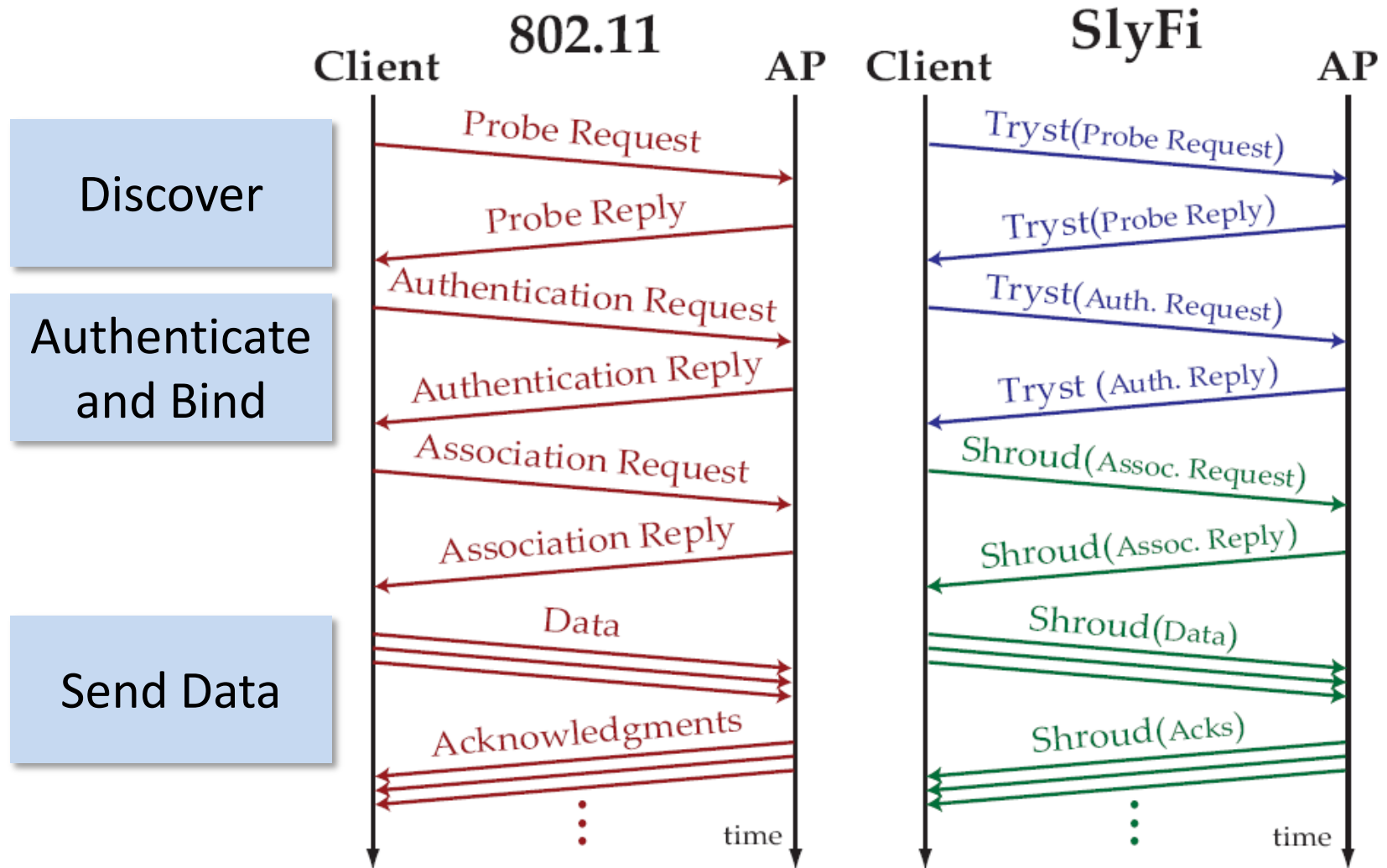
Tryst: Discovery/Binding

$header = addr_{AB}^i$	$etext = AES-CBC_{k_{s:AB}^{Enc}, header}(p)$	$emac = AES-CMAC_{k_{s:AB}^{MAC}}(header, etext)$
16 bytes	variable	16 bytes

Shroud: Data Transport

k_p $k_{AB}^{Enc}, k_{AB}^{MAC}, k_{AB}^{addr}$ $k_{s:AB}^{Enc}, k_{s:AB}^{MAC}$	<p>A one-time use key for encrypting a payload.</p> <p>Long-term keys to encrypt, MAC, and compute addresses for Tryst messages sent from A to B.</p> <p>Session keys to encrypt and MAC Shroud messages sent from A to B.</p>
--	--

Protocol Timing Diagram

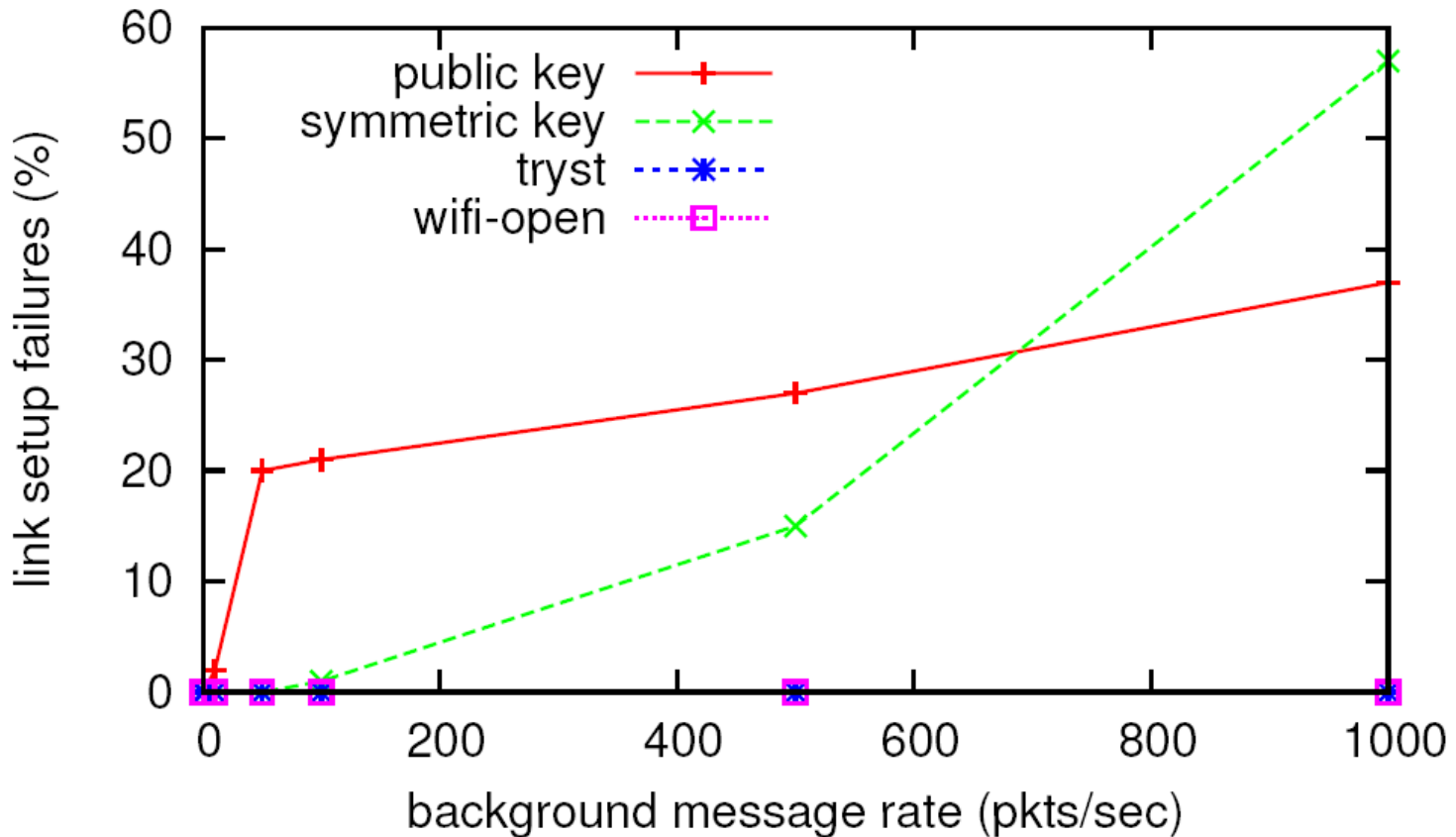


Other Protocol Details

- Broadcast
 - All broadcast packets routed through the AP
 - Use same shared key for all the clients of the AP
- Higher-layer binding
 - Clients report “pseudonym MAC address”-to-IP address bindings to AP
 - AP answers all ARP queries
- Time synchronization and roaming
 - Use protected broadcast to transmit timestamps, same BSSID info
- Coexistence with 802.11
 - Encapsulate SlyFi in “anonymous” 802.11 frame with unused FC code
 - Clients first search for SlyFi AP, then fall back to non-private AP search
- Link-layer ACKs
 - If fast enough, just acknowledge last SlyFi token sent
 - Our software implementation uses windowed ACKs

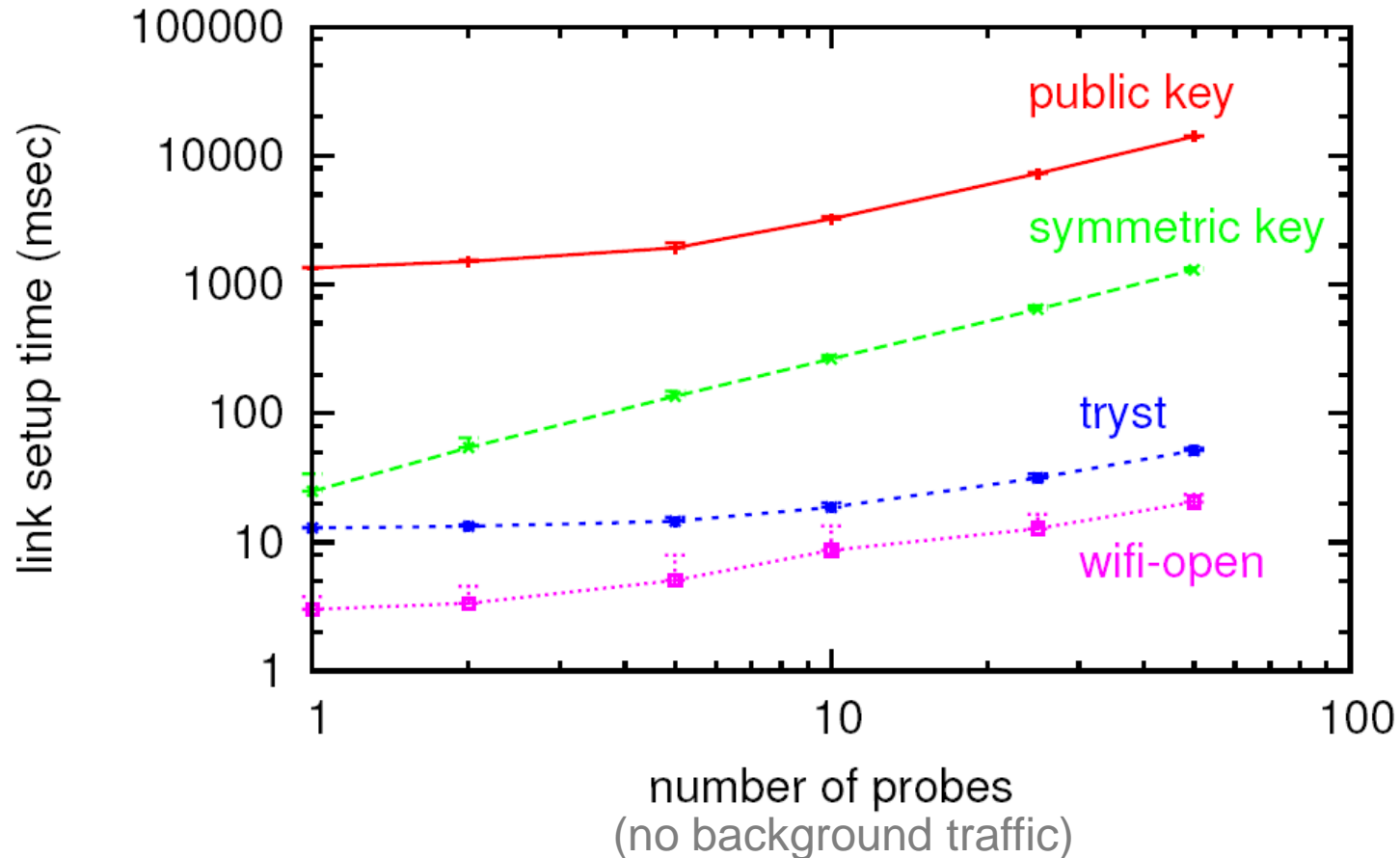
===== TRYST EVAL =====

Link Setup Failures



“Encrypt everything” fails to setup many links

Link Setup Time vs. # Probes



SlyFi scales as gracefully as 802.11

Link Setup Time Breakdown

	probing	openauth	associate	wpa-key	total
public key	886.1	895.2	146.2	NA	1927.6
symmetric key	120.2	8.6	6.9	NA	135.6
tryst	3.3	5.1	6.2	NA	14.5
wifi-open	1.4	1.5	2.2	NA	5.1
wifi-wpa	0.1	6.9	0.8	57.5	65.3

(times are in msec, no background traffic)

Using software encryption on 256 Mhz Geode processor and 802.11a

“Try all keys” dominates symmetric key time

Token Computation Time

# keys	1	10	50	100	500	1000	10,000
time (msec)	0.08	0.49	2.3	4.7	24	47	800

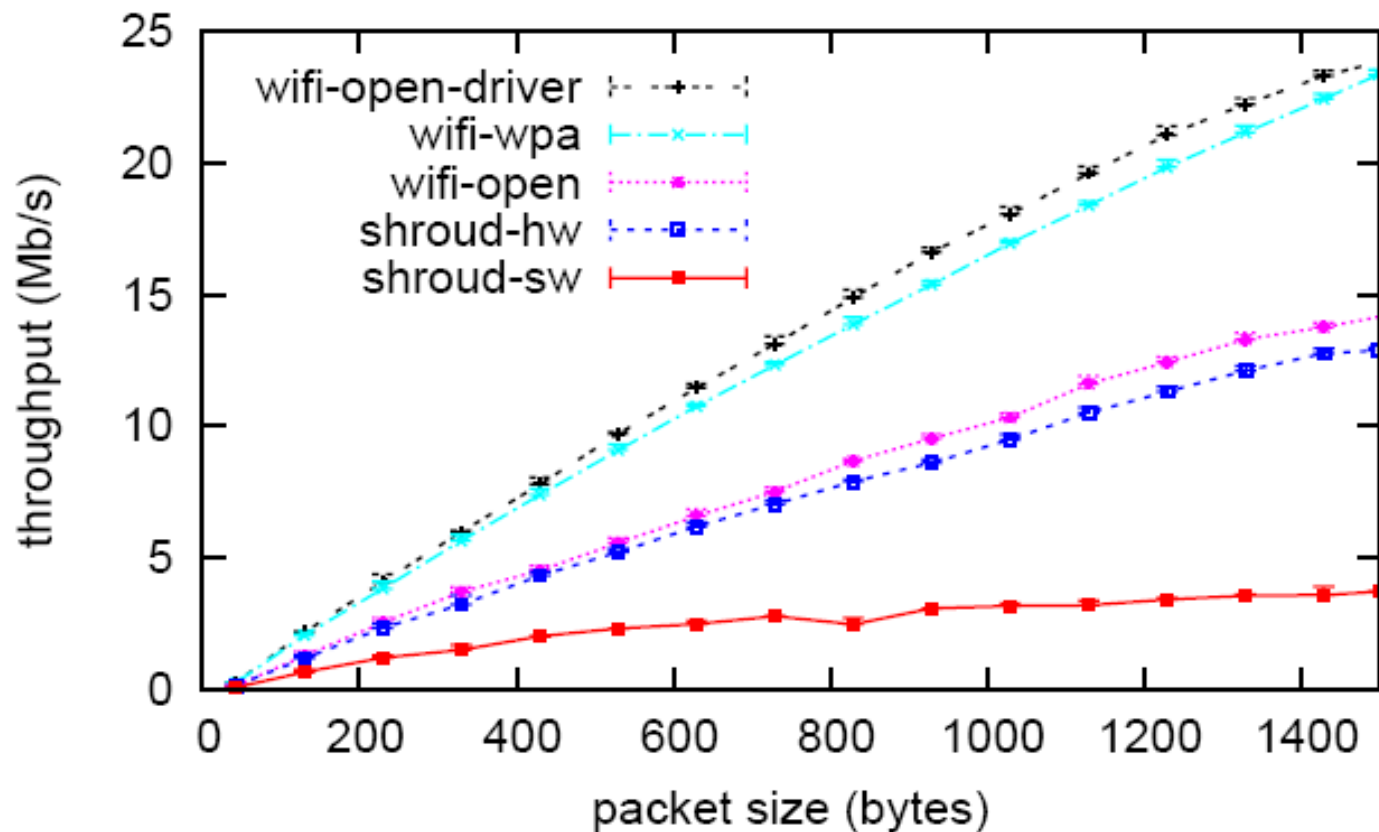
(Once every 5 minutes)

Using software AES, 256 Mhz Geode processor

Token computation time is negligible

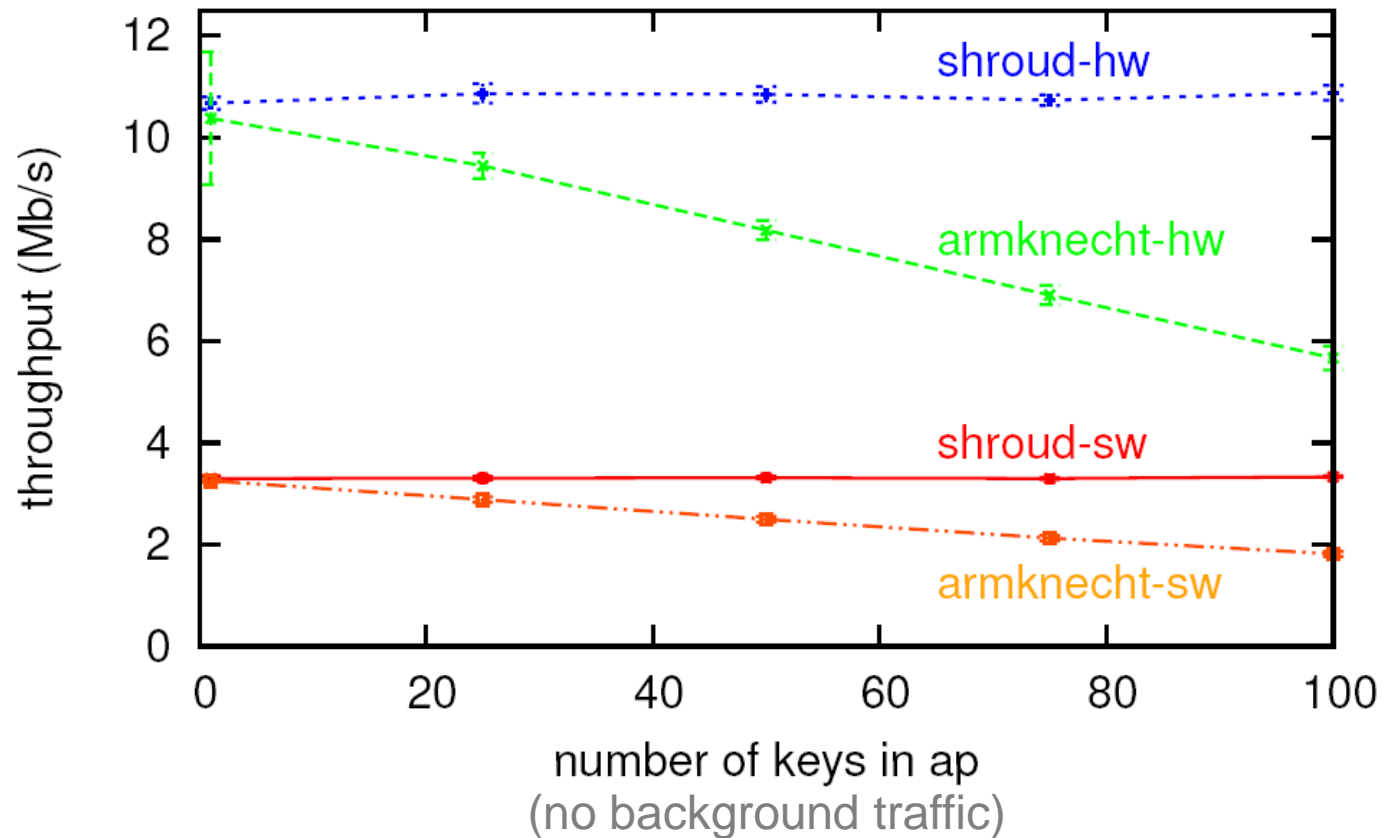
===== SHROUD EVAL =====

Data Throughput vs. Packet Size



SlyFi data transport overhead is similar to WPA

Data Throughput vs. # Associations



SlyFi throughput is independent of # associations

Data Transport Time Breakdown

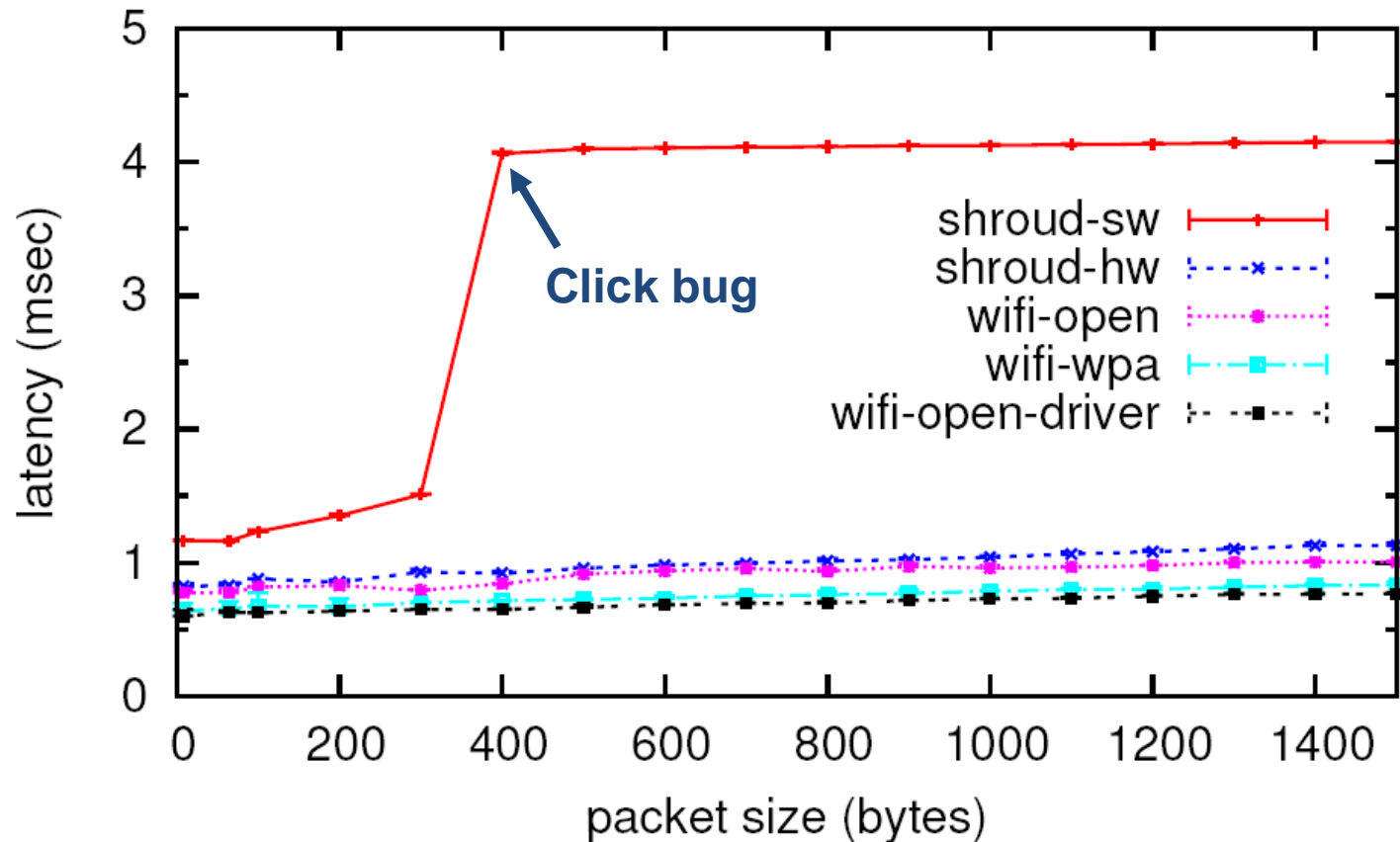
	send		filter		receive	
	sw	hw	sw	hw	sw	hw
update <i>addrs</i>						
(max message loss)	15	14	NA	NA	2047	2003 (50)
(no message loss)	15	14	NA	NA	119	117 (1)
process <i>etext</i>	951	16	NA	NA	1541	16
process <i>emac</i>	740	16	NA	NA	740	16
Shroud total	1821	120	32	32	3290	290
Click total	1913	215	144	144	3402	407

(times are in usec)

256 Mhz Geode processor, hw times based on Atheros a/b/g 802.11 card

SlyFi can process messages at the line rate

Latency vs. Packet Size



SlyFi data transport overhead is similar to WPA

===== MEASUREMENTS =====

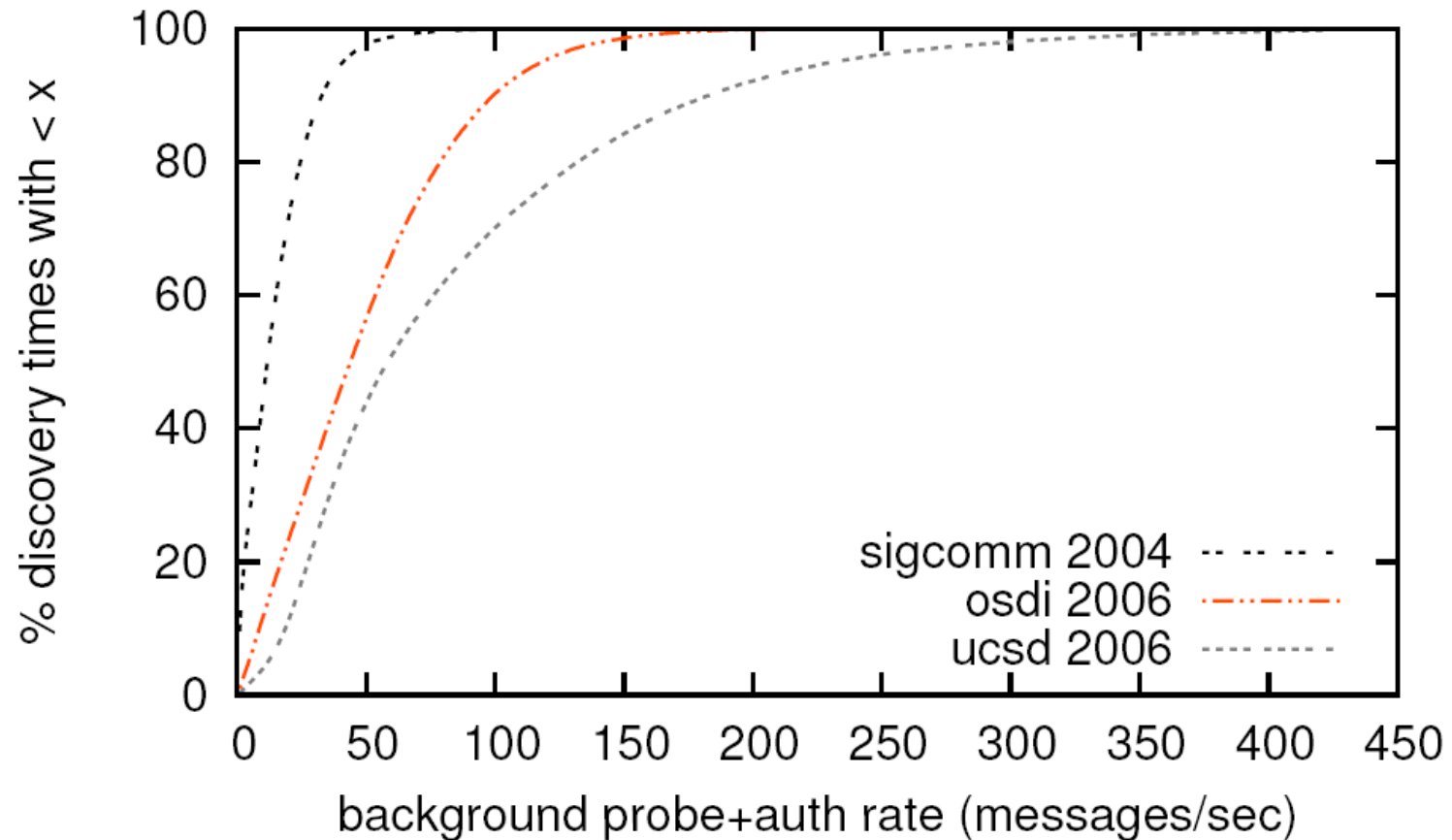
Empirical Stream Interleaving

	10 ms	100 ms	1 sec	1 min	1 hr
SIGCOMM 2004	1.4	3.2	7.6	24.7	80.1
OSDI 2006	4.6	9.0	20.6	60.8	221.3
UCSD 2006	2.4	7.1	17.9	76.6	176.6

Table 1—Mean number of devices that send or receive 802.11 data packets at different time intervals at two conferences (SIGCOMM [24], OSDI [10]) and one office building (UCSD [11]). Intervals with no data packets are ignored. UCSD has observations from multiple monitors.

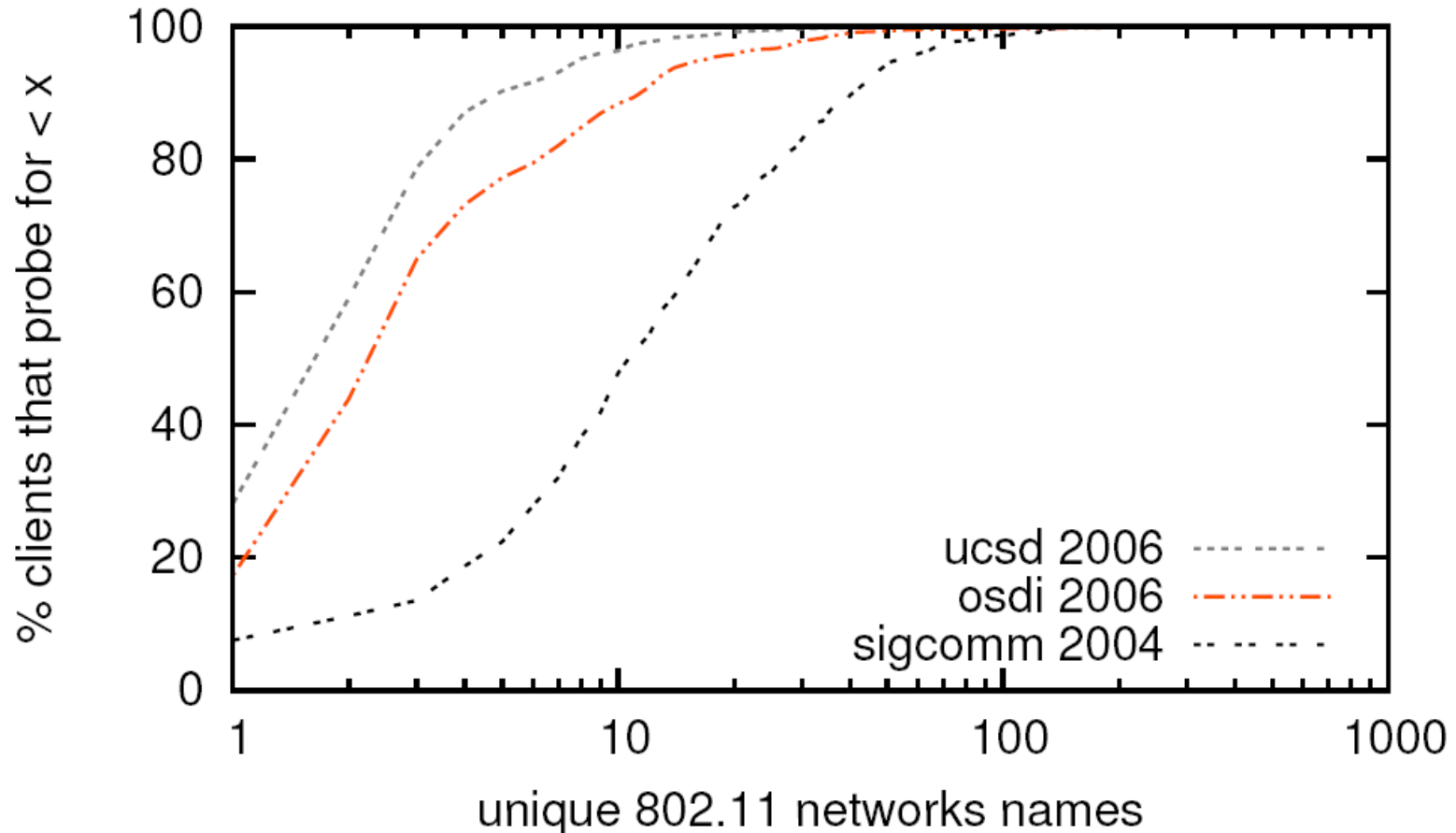
Many streams interleaved even at short timescales

Empirical Background Probe Rate



Background probes are frequent in practice

Empirical # Saved Network Names



Some clients probe for many network names