How do you Improve on the Internet?

- The Internet has been tremendously successful
  - Supports very diverse set of applications and services
  - Integral part of our society and economy
  - But there are also many challenges...
- Lots of exciting research on how to improve Internet
  - Security, routing, wireless/mobile, management, ...
  - But Internet architecture constrains what can be modified
- Future Internet Architecture frees researchers to go beyond today’s IP architecture and infrastructure
  - Multi-phase, NSF-funded research program
  - Five teams building full scale networks

“Narrow Waist” of the Internet Key to its Success

- Has allowed Internet to evolve dramatically
- But now an obstacle to addressing challenges:
  - No built-in security
  - New usage models a challenge
  - Limited interactions edge-core
- XIA exploring three concepts to address issues:
  - Diverse types of end-points
  - Intrinsic security
  - Flexible addressing
Outline

- Background
- XIA principles
- XIA architecture
- Building XIA
- Ongoing research
- Conclusion

XIA Vision

We envision a future Internet that:

- Is trustworthy
  - Security broadly defined is the biggest challenge
- Supports long-term evolution of usage models
  - Including host-host, content retrieval, services, ...
- Supports long term technology evolution
  - Not just for link technologies, but also for storage and computing capabilities in the network and end-points
- Allows all actors to operate effectively
  - Despite differences in roles, goals and incentives

Today’s Internet

- Client retrieves document from a specific web server
  - But client mostly cares about correctness of content, timeliness
  - Specific server, file name, etc. are not of interest
- Transfer is between wrong principals
  - What if the server fails?
  - Optimizing transfer using local caches is hard
    - Need to use application-specific overlay or transparent proxy – bad!

eXpressive Internet Architecture

- Client expresses communication intent for content explicitly
  - Network uses content identifier to retrieve content from appropriate location
- How does client know the content is correct?
  - Intrinsic security! Verify content using self-certifying id: 
    hash(content) = content id
- How does source know it is talking to the right client?
  - Intrinsic security! Self-certifying host identifiers
A Bit More Detail ...

Flexible Trust Management

Diverse Communicating Entities

Intrinsic Security

Where

Dest: Service ID
Content Name?

Dest: Client ID
Content ID

Dest: Content ID

Hash( ) = CID?

What About Dynamic Content?

Can use other XID Types

Dest: Service ID
Content Name?

Dest: Client ID
CID/SID/…

Dest: CID/SID/…

Hash( ) = CID?

Three Key Principles

• An set of principals allows direct identification of the intended communicating entities
  – Not having to force communication at a lower level (hosts in today’s Internet) reduces complexity and overhead
• Set up principals can evolve over time
  – Adapt to changes in usage models
  – Support custom requirements of specific deployments
• Intrinsic security guarantees security properties as a direct result of the design of the system
  – Do not rely on external configurations, actions, data bases

Other XIA Principles

• Narrow waist for all principals
  – Defines the API between the principals and the network protocol mechanisms
• Narrow waist for trust management
  – Ensure that the inputs to the intrinsically secure system match the trust assumptions and intensions of the user
  – Narrow waist allows leveraging diverse mechanisms for trust management: CAs, reputation, personal, …
• All other network functions are explicit services
  – Keeps the architecture simple and easy to reason about
  – XIA provides a principal type for services (visible)

Look familiar?
XIA: eXpressive Internet Architecture

- Each communication operation expresses the intent of the operation
  - Also: explicit trust management, APIs among actors
- XIA is a single inter-network in which all principals are connected
  - Not a collection of architectures implemented through, e.g., virtualization or overlays
  - Not based on a “preferred” principal (host or content), that has to support all communication

Outline

- Background
- XIA principles
- XIA architecture
  - Multiple principals
  - DAG-based addressing
  - Intrinsic security
- Building XIA
- Ongoing research
- Conclusion

Multiple Principal Types

- Associated with different forwarding semantics
  - Support heterogeneity in usage and deployment models
  - Set of principal types can evolve over time
- Hosts XIDs support host-based communication – who?
- Service XIDs allow the network to route to possibly replicated services – what does it do?
  - LAN services access, WAN replication, ...
- Content XIDs allow network to retrieve content from “anywhere” – what is it?
  - Opportunistic caches, CDNs, ...
- Autonomous domains allow scoping, hierarchy

Multiple Principal Types

Choice involves tradeoffs:

- Control • Trust
- Efficiency • Privacy
Content-centric Optimizations

Many Alternatives!

Supporting Evolvability

- Introduction of a new principal type will be incremental – no “flag day”!
  - Not all routers and ISPs will provide support from day one
- Creates chicken and egg problem - what comes first: network support or use in applications
- Solution is to provide an intent and fallback address
  - Intent address allows in-network optimizations based on user intent
  - Fallback address is guaranteed to be reachable

Addressing Requirements

- Fallback: intent that may not be globally understood must include a backwards compatible address
  - Incremental introduction of new XID types
- Scoping: support reachability for non-globally routable XID types or XIDs
  - Needed for scalability
  - Generalize scoping based on network identifiers
  - But we do not want to give up leveraging intent
- Iterative refinement: give each XID in the hierarchy option of using intent
Our Solution: DAG-Based Addressing

• Uses direct acyclic graph (DAG)
  – Nodes: typed IDs (XID; expressive identifier)
  – Outgoing edges: possible routing choices

• Simple example: Sending a packet to HID₅

  Dummy source: special node indicating packet sender

  Intent: final destination of packet with no outgoing edges

Support for Fallbacks with DAG

• A node can have multiple outgoing edges

  Primary edges

  Fallback edge (low priority edge)

  Intermediate node

• Outgoing edges have priority among them
  – Forwarding to HID₅ is attempted if forwarding to CIDₐ is not possible – Realization of fallbacks

DAGs Support Scoping and Iterative Refinement

Client side

Server-side domain hierarchy

Raising many interesting questions!

DAG Addressing Research Questions

• DAG addressing supports is flexible ...
  – Fallback, binding, source routing, mobility, ...

• ... but many questions remain:
  – Is it expensive to process?
  – How big will the addresses be?
  – How do ISPs verify policy compliance?
  – Can they be used to attack network?
  – Can it be deployed incrementally?
Incremental Deployment of XIA

- **4ID**: IPv4 address as an XID
  - IPv4 encapsulation between XIA network islands
  - Leverages fallback for legacy networks
- No need for statically configured tunnels!

4ID in Action: Partially Deployed XIA Networks

- Entering IPv4 network: Encapsulate XIA packet with IP header
- Entering XIA network: Remove IP header for native XIA packet processing

Works for arbitrary pairs of XIA networks

4ID in Action: Fully Deployed XIA Networks

- Use native XIA forwarding and ignore fallback

Seamless incremental deployment of XIA

Intrinsic Security in XIA

- XIA uses self-certifying identifiers that guarantee security properties for communication operation
  - Host ID is a hash of its public key – accountability (AIP)
  - Content ID is a hash of the content – correctness
  - Does not rely on external configurations
- Intrinsic security is specific to the principal type
- Example: retrieve content using ...
  - Content XID: content is correct
  - Service XID: the right service provided content
  - Host XID: content was delivered from right host
Example of Secure Mobile Service Access

XIA Internet

Server S: HIDS SIDBuf
ADbuf:SIDbuf
AD2:SID2
ADC:SIDC

Client C: HIDC SIDC

Register “bof.com”

SIDBuf SIDbuf

AdBuf-HIDb:SIDbuf
ADC-HIDc:SIDC

XIA Dataplane Concepts Revisited

Multiple Communicating Principal Types

Directly support diverse network usage models

Evolution of principle types

Customization

Flexible Addressing

Deal with routing “failures”

Intrinsic Security

Principal-specific security properties

Built in security forms basis for system level security

Introducing XIA

• Three core XIA concepts can be introduced independently
  – Each provides opportunities for improvement
  – Core ideas leverage each other
• Core ideas can be realized in variety of ways
  – Different intrinsic security properties, address formats, principal types, etc.
• Ideas can first be introduced in targeted network deployments
  – Mobile access, sensor networks, smart grid, ...

Incremental Deployment - Example
SCION Architectural Goals

- High availability, even in presence of malicious parties
- Explicit trust for network operations
- Minimal TCB: limit number of entities that need to be trusted for any operation
  - Strong isolation from untrusted parties
- Operate with mutually distrusting entities
  - No single root of trust
- Enable route control for ISPs, receivers, senders
- Simplicity, efficiency, flexibility, and scalability

Hierarchical Decomposition

- Split network into a set of trust domains (TD)
  - TD: isolation of route computation
  - TD cores: interconnected large ISPs

Path Selection in SCION Architecture Overview

- Source/destination can choose among up/down hill paths
- Path control shared between ISPs, receivers, senders
- Desirable security properties:
  - High availability, even in presence of malicious parties
  - Explicit trust for operations
  - Minimal TCB: limit number of entities that must be trusted
  - No single root of trust
  - Simplicity, efficiency, flexibility, and scalability

Distributed Control in XIA

- Customers have more choices:
  - Choice of XID type, i.e. how is communication operation performed; involves different tradeoffs
  - DAGs add flexibility: fallback, services, ...
  - Scion offers some control over path selection
- Service providers have choices as well
  - Use of XID types to optimize new services
  - Scion allows new path optimization options
  - Use DAGs for binding, scoping, mobility, ...
- Provides opportunities for customizing interactions to context
**Outline**

- Background
- XIA principles
- XIA architecture
- Building XIA
  - Forwarding packets
  - Building a network
  - Prototype
- Ongoing research
- Conclusion

**Putting DAG Addresses into Packet Headers**

**Graphic view**

- Node 0
- Node 1
- CID_A
- HID_S

**Per-node view**

- Node -1
- Node 0
- HID_S
- Node 1
- Node 1
- CID_A

**XIP Packet Header**

- DAGs represent source and destination addresses
- Array of nodes with pointers
- Maintains a LastNode field in the header
  - Routers to know where to begin forwarding lookups

```
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>XIP1.0</td>
</tr>
<tr>
<td>Next Header</td>
<td></td>
</tr>
<tr>
<td>Payload length</td>
<td></td>
</tr>
<tr>
<td>Hop Limit</td>
<td></td>
</tr>
<tr>
<td>XID type</td>
<td></td>
</tr>
<tr>
<td>160 Bit ID</td>
<td></td>
</tr>
<tr>
<td>Edge 0-3</td>
<td></td>
</tr>
</tbody>
</table>

```

**Router’s View on Packet Forwarding**

- Last visited node (In packet header)
  1. Forward to SID_S if possible
  2. Otherwise, forward to AD_S
    - If router is AD_S itself, update last visited node to AD_S
Packet Processing Pipeline

• Principle-independent processing defines how to interpret the DAG
  • The core XIA architecture
• Principle-dependent processing realizes forwarding semantics for each XID type
• Optimizations possible: fast path processing, packet level and intra-packet parallelism

Evaluation Setup

• Use packet generator to evaluate throughput
• Software:
  – PacketShader I/O Engine
  – Click modular router – multithreaded(12 threads)
• Hardware:
  – 10Gbit NIC : 4 ports (multi-queue support)
  – 2x 6 Core Intel Xeon @ 2.26GHz
• Optimizations apply: fast path processing, packet level and intra-packet parallelism

Forwarding Performance Comparison

XIP forwarding is fast!
@ 128 byte FB0 is 8% slower than IP
@ 192 byte FB3 is 26% slower than IP

Fast Path Performance

Using fast-path processing, the gap between FB0 and FB3 is reduced significantly!
Summary

- XIA packet forwarding cost is reasonably competitive compared with IP!
- Inter-packet parallelism and fast-path can be applied to get high-speed XIA forwarding on software routers
- Intra-packet parallelism can be used for further speedup in hardware implementations

XIA Prototype

- Full stack for routers, caches, and end-points
- Based on Click protocol framework
  - User-level/in-kernel, native/overlay
- XIA forwarding engine was used in performance study
- Expanded to support applications, services
  - “xsocket” programming interface
  - basic transport: datagrams, streaming, content
  - Routing, naming, diagnostics, ...

XIP Protocol Stack

Open Source XIA Release

- XIA Prototype released in May 2012
  - Includes full XIA protocol stack and utilities
  - Support for GENI and VM-based experiments
  - Improve over time with research results
  - More info: http://www.cs.cmu.edu/~xia
- integrate with Scion in late summer
  - Will be based on a Scion path XID type
- Prototype good platform for collaboration
  - We can provide support to users and developers
Outline

• Background
• XIA principles
• XIA architecture
• Building XIA
• Ongoing research
• Conclusion

XIA Components and Interactions

Supporting Applications and Services over XIA

• Key to evaluating, improving the architecture
  – Goal is permanent XIA deployment
• Port simple applications to XIA
  – E.g., ftp, telnet – basic but useful
• Efficient support for content retrieval
  – What should URLs look like, dynamic content, ...
• Conferencing applications
  – Can we make use of caches, CIDs?
  – What type of multicast support is needed

Using In-Path Services

• Use XIA to better support in-path services
  – Builds on the Tapa transport architecture
• Raises research questions in many areas
  – What type of DAGs are effective and for what services?
  – How do transport protocols and services interact?
  – What are the intrinsic security properties of a session?
  – How can DAGs be safely modified during a session?
The IP Abstraction Today

Can no longer hide differences!

Wireless and Mobile Challenges

- Increasing network heterogeneity
  - Paths are no longer homogeneous
- Topology control
  - Handoff, multi-path
- Heterogeneous devices, usage
  - Relaxed end-point synchronization
- Diverse network services
  - Content retrieval, mobility services

Transfer Access Points

- Tapa supports visible middleboxes (TAPs) that break up end-end connections in homogeneous segments
- Segments support best effort delivery of “chunks”
  - Each segment can use custom solutions for congestion, flow, and error control
  - Chunks are self-certifying (ADUs)

Unbundling the Transport Layer

- Transfer layer glues segments into e-e path
  - Kind of like IP, but across segments, not hops
  - Naturally supports insertion of network services
- Thin end-to-end transport supports e-e semantics
  - Also flow, error, congestion control across segment path
  - Must account for failures of TAPs, segment breaks, etc.
Vehicular Example

- Vehicle-infrastructure suffers from frequent interruptions, short periods of connectivity
- Vehicle optimizes transfers by explicitly managing server-TAP and TAP-vehicle transfers
  - Leverages self-certifying content identifiers

Bandwidth Discrepancy in End-to-end Transfers

- Catnap uses this opportunity to save energy
- TAP buffers incoming packets while client sleeps
- Scheduler schedules burst transfer to maximize energy savings while avoiding e-e delay
  - Estimates bandwidth in wired and wireless segments

How Much Can the NIC Sleep?

- TCP transfers remain in active state
- Transfer times do not increase with Catnap
- Sleep time with Catnap increases as transfer size increases

Tapa and XIA

- Content-centric optimizations in Tapa can be pushed “into the network”
  - Tapa can use content XIDs rather than host XIDs
  - Old APs can be listed as hints (rather than server)
- Tapa needs support from services on/near APs
  - Simple “decoupling services”, content optimization, Catnap, higher level services
- Tapa benefits from intrinsic security properties
Scion over XIA Dataplan

- Store paths generated by Scion into a new type of XID
  - Sequence of MACs
  - Can be combined with other principal types
- XIA network supports both path and destination-based forwarding

How Does HTTP Chunking Work?

- **NOT** all contents are the same
- Video is fundamentally different from transaction traffic
  - Follow the traffic: 60% Internet traffic today, will be more than 95% in the next 2-3 years
  - Premium video on big screens → zero tolerance for poor quality
- XIA provides the extensible and evolvable framework to deploy mechanisms to optimize the dominant traffic type

2006 – 2011: Internet Video Going Prime Time

- Video over XIA
A Broad Research Agenda

- Applications and services
  - Web, CDNs, video delivery, teleconferencing, games, mobility services, ...
- Protocols and network infrastructure
  - Security, transport protocols, naming, mobility, routing, service deployment, principal types, network operations, diagnostics, ..
  - Optimize XIA forwarding, services, caching, ...
- Targeted deployments
  - Use XIA to optimize unique networks, e.g. wireless access, Scada, sensors, “ad hoc”, data center, ...

Three Concepts for High Quality Video Delivery

- Continuous measurement and optimization
- Multi-bit rate streams delivered using multiple CDNs
- Optimization algorithms based on
  - Individual client, and
  - Aggregate statistics at multiple time scales

Conclusion

- XIA supports evolution, expressiveness, and trustworthy operation.
  - Multiple principal types, intrinsic security, and flexible addressing
  - Open source prototype available online: http://www.cs.cmu.edu/~xia
- Looking for collaborators on broad research agenda applications, protocols, and deployments
  - Use XIA to fundamentally improve the network: transport protocols, trust management, applications, services, ...
  - Use flexibility to target demanding network deployments
    - Customize without giving up interoperability