Cryptogaphy and the Internet

Steven M. Bellovin

smb@research.att.com

973-360-8656

AT&T Labs Research

Florham Park, NJ 07932

The Drive for Cryptography

- Internet commerce
  - People *perceive* the Internet as insecure.
  - People *perceive* cryptography as the answer.
  - Both points, of course, are partially valid.
- The Internet is *the* data network.
  - Link to branch offices.
  - Telecommute via local ISP.
  - Talk to customers, partners, vendors.
- The technology is ready.

Cryptographic Technology

- CPUs are fast enough — usually — that the overhead is tolerable.
- Many essential pieces have been standardized.
- Most — but not all — of the necessary science exists.

Current Uses of Cryptography

- Email.
- SSL — the Secure Socket Layer for the Web.
- IPSEC — network-layer encryption.
- SET — secure electronic payments.

Secure Email

- Two different schemes, PGP and S/MIME, have wide penetration.
- Both appear to be secure designs.
- But both are being changed as part of the IETF standards process.
- Both are hampered by the lack of a widespread public-key infrastructure.
  $\Rightarrow$PGP's Web of Trust doesn't scale to very large populations.

Secure Socket Layer

- General mechanism used almost exclusively for user-to-Web server traffic.
  ⇒Used only for purchases; rarely used to hide browsing patterns.
- Servers have certificates; clients can, but almost never do.
- Users rarely check certificates — they don't know what certificates are, who has signed them, or when they should or should not be accepted.
- In other words, we have a Web PKI, but it's effectively unused.

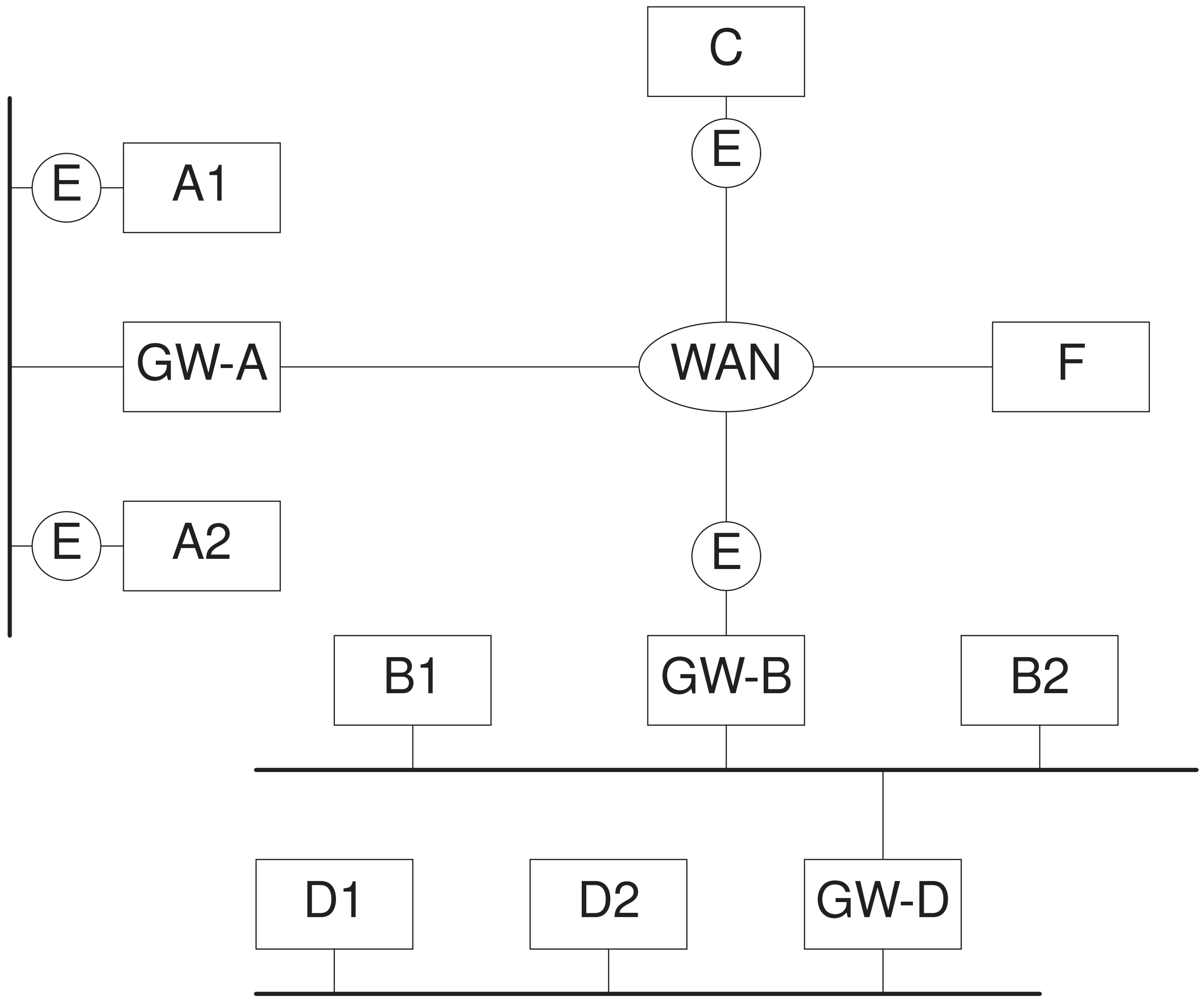
IPSEC

- IPSEC operates at the “network layer”.
  - Protects all transport protocols.
  - Protects all applications that use the transport protocols.
- Possible to trade cost for granularity of protection — one key (and IPSEC module) can protect a user, a host, or an entire network.

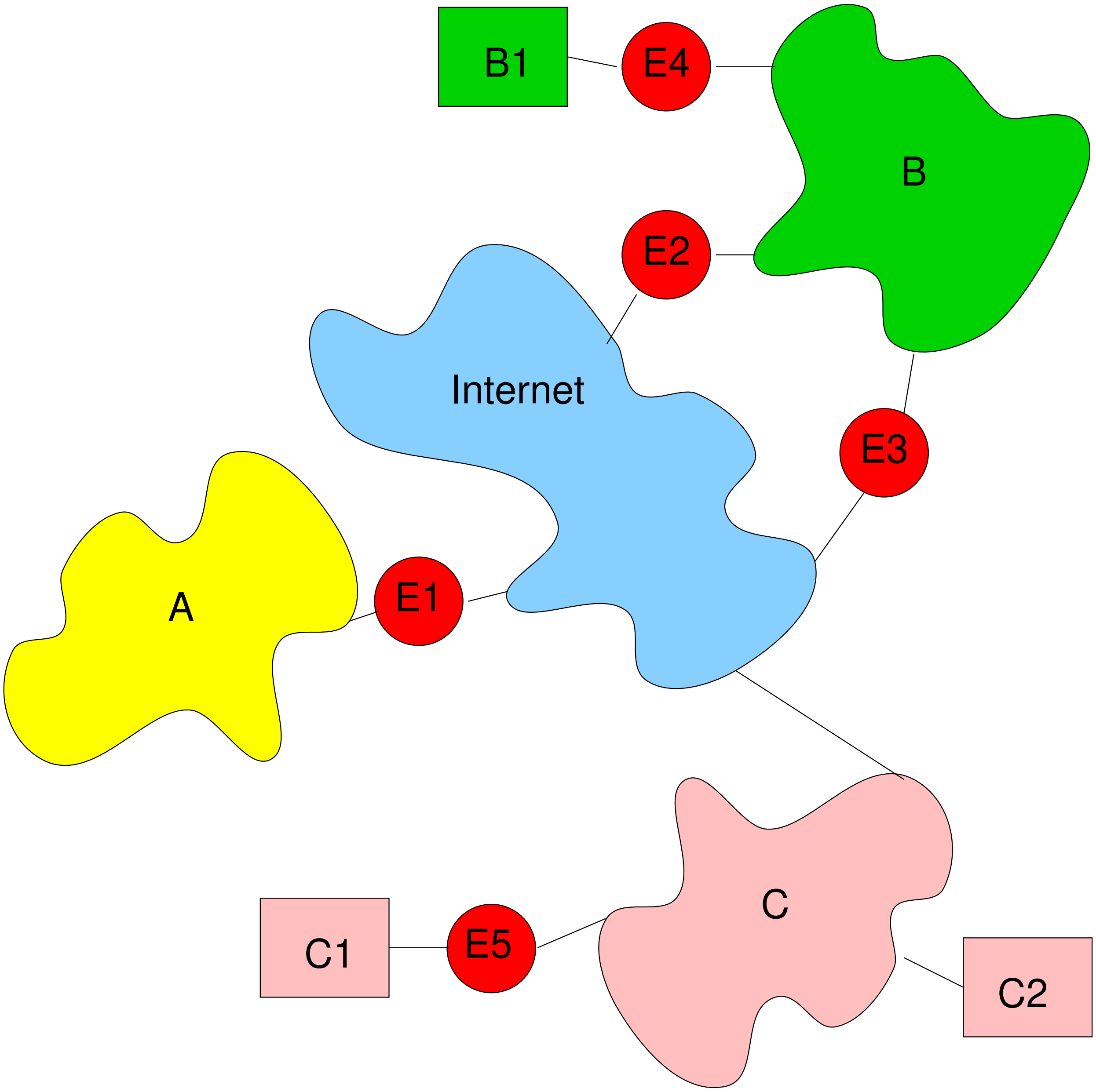
IPSEC Deployment Patterns

- Initial uses likely to be firewall-to-firewall and user-to-firewall.
- Certificates will often connote authorization for firewall traversal; this implies that a generic PKI won't be needed at first.
- Windows NT 5.0 will (probably) include IPSEC; that may lead to much more end-system use of IPSEC, which in turn will require a PKI.

IPSEC Patterns



Possible Topology



Finding Gateways

- How does a host on net C know to use E1 to reach net A?
- If E3 goes down when C is talking to B, can the connection switch to E2? Can E2 and E3 share the load? What if a topology change makes a different gateway "better"?
- Who signs the delegation? The end-host? The firewall administrator?
- If C1 wishes to talk to B1, how does it know to negotiate with two different IPSEC layers?
- How do you know that encryption is needed at all? How do you know — securely — that it isn't needed?
- When can you/should you talk in the clear?

SET

- A secure payment protocol — digitally sign orders, instead of keeping credit card numbers lying around the server.
- A multiparty protocol — consumer, bank, merchant.
- Credit card numbers often sent to merchant anyway — real-world usage dictates that vendors key their databases by user credit card number.

⇒ Much more complexity; little additional security. (Non-repudiation doesn't mean much in the U.S., given the requirements of U.S. law on credit card charges.)

Missing Pieces

- Speed of public key operations, especially for servers.
  ⇒Moore's Law means we ask servers to handle more clients, not do more cryptography.
- Speed of keyed integrity-check algorithms. (During IPSEC's development, we discovered many different varieties of cut-and-paste attacks.) Current algorithms appear to be poorly suited for hardware acceleration.
- Secure routing protocols.
- Secure multicast.

Secure Routing

- Internet routers exchange reachability information with each other.
- A router knows how to reach its directly-connected nets, plus those it has learned of from its neighbors. There is no global routing or topology database.
- We can protect the pairwise links easily enough, but. . .
- If a router lies, its neighbors will be deceived. A remote router has no way of knowing what's going on.
- We may need digital signature chains back to the (authorized) origin. But that's too expensive.
- Besides, many routes are plausible; which is *currently* correct?

Secure Multicast

- Many different models for multicast: broadcast, broadcast plus Q&A, private conversation.
- Trust models in the literature are often wrong — in the Internet, the key distribution graph is often not equivalent to the packet-forwarding graph.

Trust Management

- We already discussed IPSEC encryption policies and gateways.
- What is the relationship between the real world and the Internet? Can `interactive.wsj.com` use a certificate for `www.wsj.com`? The certificate is owned by Dow Jones; is that right? What about `nasa.gov` versus `nasa.com`?
- What are the valid uses for a certificate? Is the same certificate good for both the Web and email to customer care?
- Internet certificates will be used by programs, not people; how do we automate the semantic validation?

Cryptography vs. Cryptographic Engineering

- Cryptography deals with abstract notions of message exchanges.
- Cryptographic engineering specifies these messages well enough that someone can implement them.
- It also deals with the dichotomy (and occasionally tension) between cryptography and the real, external environment.

Encrypting a Message

- An academic paper says "A $\rightarrow$ B: $\{M\}_K$.
- A specification describes which cipher to use, what block size, how padding is to be done for the block cipher, where the IV and key come from, how long the key can be used, what to do if the message is not acknowledged, etc.
- The real world also has to accomodate different ciphers, and secure negotiation of which should be used.

Chosen Plaintext Attacks

- In the Internet, chosen plaintext attacks are often feasbile.
- Example: send a long email message to a mail server; watch as an IPSEC user downloads it.

Requirements in Conflict

- In one paper on IPSEC, I showed that fine-grained keying was preferable. In another paper, I showed that fine-grained keying aids cryptanalysts and traffic analysts. Which choice is right?
- Secure DNS was designed so that signatures could be done offline, so that the private signing key is not at risk. That causes trouble for negative answers and for dynamic DNS updates.
- Secure DNS also suffers because the existing DNS protocols transmit time-to-live fields, rather than absolute expiration times. Given the caching structure, a time-to-live field cannot be protected, thus permitting some attacks.
- Traffic engineers need to know what protocols are used, and what packet sizes are like. IPSEC hides both.
- End-to-end IPSEC conflicts with firewalls; the latter are charged with examining traffic to ensure that it is safe, but the firewall doesn't have the key.

More IPSEC Conflicts

- Some sites use “Network Address Translators”, which tinker with message addresses. They also need to look inside packets for addresses in the payload.
- People building satellite gateways want to increase the “window size”, to improve throughput.
- Wireless nets are much lossier than wired ones; some gateways like to resend packets they know you’ve already seen.
- These are *authorized* man-in-the-middle attacks. . .

Protocol Verification

- Verifying cryptographic protocols is hard enough. Verifying real-world standards is worse, because of non-cryptographic features.
- Example: bind to a “port” after your enemy’s conversation is through, and reinject the ciphertext packets into the network. The kernel will decrypt them and pass you the plaintext.
- Example: Wagner’s short-block guessing attack. With 2^8 blocks of chosen plaintext and a 2^8 packet active attack, an enemy can read certain classes of traffic, by watching for TCP acknowledgment messages.
- Implementation bugs, such as bad random number generators. Protocols with stronger requirements for random numbers (i.e., DSS) may be less appropriate.

Theoretical Help

- IPSEC originally used $H(K, \text{packet}, K)$ as an integrity check. We learned that that was not secure enough, so we switched to HMAC.
- But there was a lot of resentment at this — not because it was wrong, or from the wrong people, but because it was late.
- "Shoot the engineers and ship the product" is a common phrase; sometimes it's too late to change things easily.

What Cryptographic Science Can't Fix

- No more than 15% of CERT advisories could have been prevented by cryptography. Most of the problems were due to buggy code, sometimes in cryptographic modules.
- There's too much bad cryptography out there — bad (and home-grown) algorithms, inappropriate modes of operation, misuse of stream ciphers, pseudo-one-time pads (but with claims of theoretical security), etc.
- 40-bit keys. . .