
CS 5430

Protocol Design

Prof. Clarkson
Spring 2017

Review

- **Cryptography:** Encryption, block ciphers, block cipher modes, MACs, cryptographic hash functions, digital signatures, authenticated encryption, key derivation, key establishment, secure channel
- Today:
 - Cryptographic protocols:
 - Review of attacks
 - Design principles
 - Time permitting, case study of cryptography and electronic voting

ATTACKS ON PROTOCOLS

Attacks

- Compare:
 - "this protocol resists the following list of attacks"
 - vs.
 - "this protocol achieves the following security goals under the following assumptions"
- Both establish trustworthiness of protocol
 - latter is more useful to user of protocol
 - former is nonetheless useful to us as analysts

Attacks

- **Eavesdropping:** passively capture messages
 - primary countermeasure: encryption
- **Replay:** record and resend messages
 - maybe to same or different principal
 - maybe in same or different protocol run
 - primary countermeasure: nonces (counters, timestamps, random numbers)
 - special cases:
 - **Preplay:** attacker engages in early protocol runs with goal of attacking later run
 - **Reflection:** send protocol messages back to principal who originally sent them, often in parallel runs with flipped roles
 - **Man in the middle:** attacker interposes between two principals, perhaps pretending to be the other to each of them

Attacks

- **Modification:** actively alter messages
 - many attacks don't alter fields of message but **splice** together fields from separate messages
 - primary countermeasure: MACs, which must tie together fields to prevent splicing
 - **Typing attack:** cause principal to mis-parse message, e.g. interpret a principal identifier as a key or v.v.
- **Protocol:** attacker can run whatever protocol it wants
 - maybe another protocol that uses the same keys in a different way (which might violate our principle of using keys for unique purposes)
 - maybe a custom-designed protocol

PRINCIPLES FOR PROTOCOLS

Design principles

[Abadi and Needham 1995]

- Wisdom derived from analysis of many protocols and attacks
- Not **sufficient** to guarantee security
- Not **necessary** to guarantee security
- But following principles would have prevented mistakes

Say what you mean

Main principle: Every message should say what it means

- Interpretation of message should depend only upon content of message
- Hence recipient can recover meaning without needing to assume or supply any context
- Writing down a straightforward English sentence describing the meaning of each step in narration is good practice

Say what you mean

Protocol narrations sometimes work against this principle:

Example: 4 . B -> A: X

- Protocol designer intended...
 - it's the fourth message sent
 - the contents are X
 - B originates it
 - A receives it
- Because of attacker, none of those is necessarily true

Say what you mean

Protocol narrations sometimes work against this principle:

Another example: $S \rightarrow A: \text{Enc}(B, k_{AB}; k_{AS})$

- Might mean "*S sends to A a session key k_{AB} intended to be good for conversation with B*"
- But the narration itself doesn't say that clearly
- And if it were $S \rightarrow A: \text{Enc}(k_{AB}; k_{AS})$, then A would have to guess that the key is for B, or assume it from context of other messages in protocol

Say what you mean

Three forms of confusion: current message expected by principal vs...

- same message from previous run of same protocol
- different message of same protocol
- message from different protocol

Say what you mean

Principle: Message contents should describe what protocol, which instance, and message number in it

Example (back to Needham-Schroeder), instead of:

- 4. B→A: AuthEnc (nB; kAB)
- 5. A→B: AuthEnc (nB-1; kAB)

could verbosely use:

- 4. B→A: AuthEnc ("NS4", A, B, nB; kAB)
- 5. A→B: AuthEnc ("NS5", A, B, nB-1; kAB)

Naming

Principle: Explicitly name the relevant principals in each message

- If principals are not named, recipient has to make assumptions from context
- Assumptions are vulnerabilities
- Attacker will exploit with replay, modification attacks
- E.g., Lowe's attack on Needham-Schroeder

Naming

Another example [Denning and Sacco 1981]:

1. $A \rightarrow B: \text{Enc}(k_{AB}, t_A, \text{Sign}(k_{AB}, t_A; k_A); K_B)$

Intended meaning might be *"At time t_A , principal A says that k_{AB} is a good key for communication between A and B"*

- But the message doesn't name A or B
- Maybe it's okay not to name A, since A's private key is used
- But there's an attack that's possible because B is not named...

Naming

M gets A to start a protocol run...

1. A \rightarrow M: $\text{Enc}(k_{AM}, t_A, \text{Sign}(k_{AM}, t_A; k_A); K_M)$

Then M pretends to be A to B...

1'. M \rightarrow B: $\text{Enc}(k_{AM}, t_A, \text{Sign}(k_{AM}, t_A; k_A); K_B)$

And now maybe B discloses secrets to M, or mistakenly trusts information as having come from A, etc.

Naming

Intended meaning: "At time t_A , principal A says that k_{AB} is a good key for communication *between A and B*"

Improved protocol:

1. A \rightarrow B:

$\text{Enc}(A, B, k_{AB}, t_A, \text{Sign}(A, B, k_{AB}, t_A; k_A); K_B)$

Cryptography

Principle: Be clear about what cryptographic primitives are being used, and why, and what properties of them are needed

- Do you need confidentiality?
 - How strong does it need to be?
 - Who should be allowed to learn secrets?
 - What algorithms are acceptable? Are any unacceptable?
- Do you need integrity? (similar questions)
- Do you need both?

Cryptography

"There is considerable confusion about the uses and meaning of encryption" [Abadi & Needham]

- Sometimes (correctly) used for **confidentiality**
- Sometimes used **incorrectly for integrity**
 - Sometimes used incorrectly to bind parts of messages, i.e., prevent splicing
 - But $\text{Enc}(X,Y)$ might turn out to be exactly the same as $\text{Enc}(X),\text{Enc}(Y)$, depending on the exact Enc in use
- **Confusing notation in literature: $\{m\}_k$**
 - Sometimes used to unify notions of $\text{Enc}(m; k)$ and $\text{MAC/Sign}(m; k)$
 - Then hard to discern what properties the protocol designer wanted of that primitive

Cryptography

Principle: A principal who signs a message that is already encrypted can't be assumed to know the plaintext of that message

- From $\text{Sign}(\text{"I like ice cream"}; k_A)$, reasonable to conclude A claims to like ice cream
- From $\text{Sign}(\text{Enc}(\text{"I like ice cream"}; k); k_A)$, not reasonable to conclude that fact, because A might not have access to k

Cryptography

ISO/IEC 11770-3 Key Transport Mechanism 2:

1. $A \rightarrow B: B, t_A, \text{Enc}(A, k_{AB}; K_B), \text{Sign}(B, t_A, \text{Enc}(A, k_{AB}; K_B); k_A)$

Nothing guarantees that A actually knows the session key k_{AB}

- $\text{Enc}(A, k_{AB}; K_B)$ could have been given to A by the attacker
- So protocol does not provide **key confirmation**
- B must trust A not to sign unknown keys
(or, maybe, trust that if A does so, anyone else who knows the key is at least as trustable as A)

Cryptography

A similar issue:

1. **A**->**B**: $\text{Enc}(m; K_B)$, $\text{Sign}(m; k_A)$

which, recall, almost always practically means:

1. **A**->**B**: $\text{Enc}(m; K_B)$, $\text{Sign}(H(m); k_A)$

Nothing guarantees that A actually knows m

- $\text{Enc}(m; K_B)$ and $H(m)$ could have been given to A by the attacker
- So the protocol does not guarantee plaintext knowledge

Cryptography

Moral of the signing story:

- Be wary if a protocol ever asks a principal to sign something that is already encrypted or hashed
- Be wary if a protocol ever asks a principal to sign something that was received from someone else

Freshness

Principle: Be clear what properties are assumed of nonces

- unique? unpredictable?
- counters can guarantee uniqueness, not unpredictability
- predictable nonces are subject to replay or preplay

Freshness

Principle: Don't use nonces in place of names

- make principles restate their names for clarity of message, not just present a nonce that supposedly only they would know

Freshness

Principle: If timestamps are used as nonces, then:

1. The difference between local clocks must be much less than the allowable age of a message
2. The time synchronization mechanism becomes part of the TCB

Freshness

Principle: A key that has been used recently might be old and compromised

- as seen in replay attacks in previous lecture

Trust

Principle: State what trust assumptions are necessary, and why

Examples:

- Server must be trusted to issue correct timestamps
- Principal must be trusted to choose good keys

...applies to all of computer security!

ELECTRONIC VOTING

Electronic voting

(We talked a bit about the history of e-voting technologies and their confidentiality and integrity goals. For a full set of slides see here:

<http://www.cs.cornell.edu/courses/cs5430/2016sp/l/28-evoting/lec.pdf>. We covered the first 12.)

Upcoming events

- [today] make sure you've decrypted A3
- [next Wed] A3 due

Anyone who considers protocol unimportant has never dealt with a cat. – Robert A. Heinlein