Off-the-Record Messaging

Ian Goldberg
University of Waterloo

7th Annual Privacy and Security Workshop &
15th CACR Information Security Workshop
2 November, 2006

The challenge

Researchers have had a hard time getting their work in security and privacy technologies to benefit real people.

- It's hard to use!
- It's hard to get!
- It doesn't work!

The goal

- At the end of the day, what matters is that the technologies we produce actually improve people's lives in some way!
- Our goal is to create what we call **Useful Security and Privacy Technologies**.

Useful Security and Privacy

- There are four major aspects to such technologies:
  - Usability
  - Deployability
  - Effectiveness
  - Robustness
- We'll quickly look at what these all mean.

Usability

- Usability is the best known of these properties.
- We not only mean it in the sense of user interfaces, and “usable security”, however.
- For example, if a privacy technology causes your web browsing to slow to an unacceptable crawl, that's an unusable technology.

Deployability

- But making a technology easy to use isn't enough!
- It also has to be *reasonable* to use.
  - If users have to change their:
    - operating systems
    - web browsers
    - instant messaging clients
  - then they won't want to use your technology.

Effectiveness

- Of course, even assuming the users *have* the technology, it needs to do them some good.
- All too often, we see that many proposed, and even widely deployed, security systems have major flaws.
  - Peer review, analysis
  - Not only of the design, but also the implementation

Robustness

- Many times, security technologies work only so long as everything goes “according to plan”.
  - Small deviations from the assumptions made by designers can cause the systems to fail catastrophically!
- But:
  - Users forget passwords
  - Their computers are compromised by malware
  - They misunderstand security-relevant messages
  - They fall victim to phishing attacks
  - etc.

An example

- Alice and Bob want to communicate privately over the Internet.

- Generous assumptions:
  - They both know how to use PGP
  - They both know each other's public keys
  - They don't want to hide the *fact* that they talked, just what they talked about

Solved problem

- Alice uses her private signature key to sign a message
  - Bob needs to know who he's talking to
- She then uses Bob's public key to encrypt it
  - No one other than Bob can read the message
- Bob decrypts it and verifies the signature

- Pretty Good, no?

Plot Twist

- Bob's computer is stolen by “bad guys”
  - Criminals
  - Competitors
  - Subpoenaed by the RCMP
- Or just broken into
  - Virus, trojan, spyware, etc.
- **All** of Bob's key material is discovered
  - Oh, no!

The Bad Guys Can...

- Decrypt past messages
- Learn their content
- Learn that Alice sent them
- And have a mathematical **proof** they can show to anyone else!

- How private is that?

What went wrong?

- Bob's computer got stolen?
- How many of you have never...
  - Left your laptop unattended?
  - Not installed the latest patches?
  - Run software with a remotely exploitable bug?
- What about your friends?

What Really Went Wrong

- PGP creates lots of incriminating records:
  - Key material that decrypts data sent over the public Internet
  - Signatures with proofs of who said what
- Alice had better watch what she says!
  - Her privacy depends on Bob's actions

Casual Conversations

- Alice and Bob talk in a room
- No one else can hear
  - Unless being recorded
- No one else knows what they say
  - Unless Alice or Bob tells them
- No one can **prove** what was said
  - Not even Alice or Bob
- These conversations are “off-the-record”

We Like Off-the-Record Conversations

- Legal support for having them
  - Illegal to record conversations without notification
- We can have them over the phone
  - Illegal to tap phone lines
- But what about over the Internet?

Crypto Tools

- We have the tools to do this
  - We've just been using the wrong ones
  - (when we've been using crypto at all)

- We want **perfect forward secrecy**

- We want **deniable authentication**

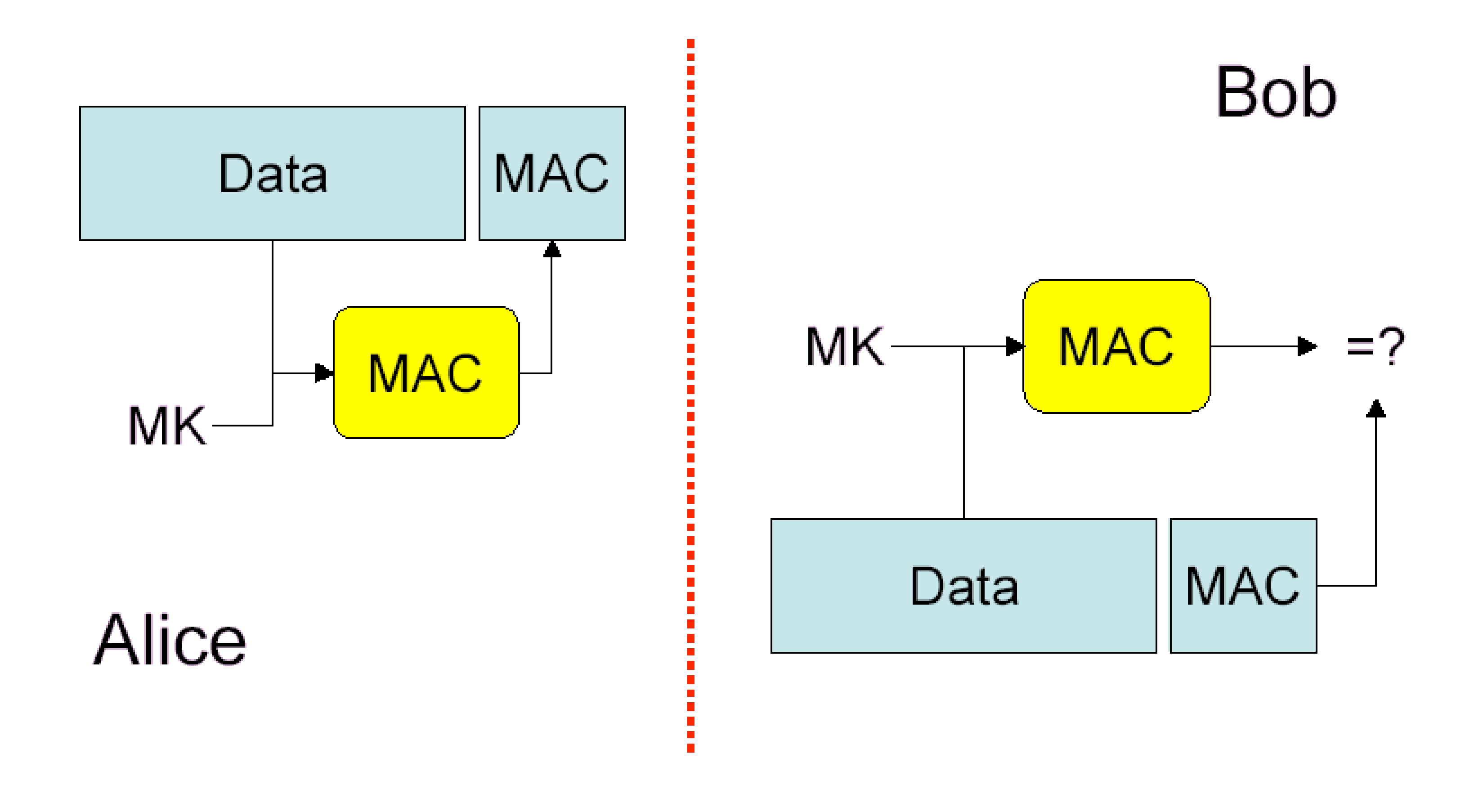
Perfect Forward Secrecy

- Future key compromises should not reveal past communication
- Use a short-lived encryption key
- Discard it after use
  - Securely erase it from memory
- Use long-term keys to help distribute and authenticate the short-lived key

Deniable Authentication

- Do **not** want digital signatures
  - Non-repudiation is great for signing contracts, but undesirable for private conversations
- But we **do** want authentication
  - We can't maintain privacy if attackers can impersonate our friends
- Use **Message Authentication Codes** (MACs)

MAC Operation



No Third-Party Proofs

- Shared-key authentication
  - Alice and Bob have the same MK
  - MK is required to compute the MAC
- Bob cannot prove that Alice generated the MAC
  - He could have done it, too
  - Anyone who can verify can also forge
- This gives Alice a measure of deniability

Using these techniques

- Using these techniques, we can make our online conversations more like face-to-face “off-the-record” conversations
- But there's a wrinkle:
  - These techniques require the parties to communicate *interactively*
  - This makes them unsuitable for email
  - But they're still great for instant messaging!

Off-the-Record Messaging

- Off-the-Record Messaging (OTR) is software that allows you to have private conversations over instant messaging, providing:
- Encryption
  - Only Bob can read the messages Alice sends him
- Authentication
  - Bob is assured the messages came from Alice

Off-the-Record Messaging

- Perfect Forward Secrecy
  - Shortly after Bob receives the message, it becomes unreadable to anyone, anywhere
- Deniability
  - Although Bob is assured that the message came from Alice, he can't convince Charlie of that fact
  - Also, Charlie can create *forged transcripts* of conversations that are every bit as accurate as the real thing

Off-the-Record Messaging

- Availability of OTR:
  - It's built in to Adium X (a popular IM client for OSX)
  - It's a plugin for gaim (a popular IM client for Windows, Linux, and others)
    - With these two methods, OTR works over almost any IM network (AIM, ICQ, Yahoo, MSN, etc.)
  - It's a proxy for other Windows or OSX AIM clients
    - Trillian, iChat, etc.
  - Third parties have written plugins for other IM clients
    - Miranda, Trillian

Is OTR Useful?

- OTR is easy to use
  - The software automatically notices when Alice and Bob both support OTR, and automatically protects their conversations.
  - The IM servers just pass encrypted messages back and forth between Alice and Bob, unaware that anything unusual is going on.

Is OTR Useful?

- OTR is easy to deploy
  - You probably don't have to change your IM client to use OTR.
  - In fact, your IM client might support OTR already!
  - It's also part of many standard OS distributions.

Is OTR Useful?

- It works
  - Peer-reviewed design
  - Open-source implementation
- Robust against failures
  - Preserves security in the face of simple failures
  - Preserves deniability in the face of major failures

Conclusion

- OTR is a good example of a Useful Security and Privacy Technology.
- Tens of thousands of people are using OTR to protect their IM conversations.
- More information at:

http://otr.cypherpunks.ca/