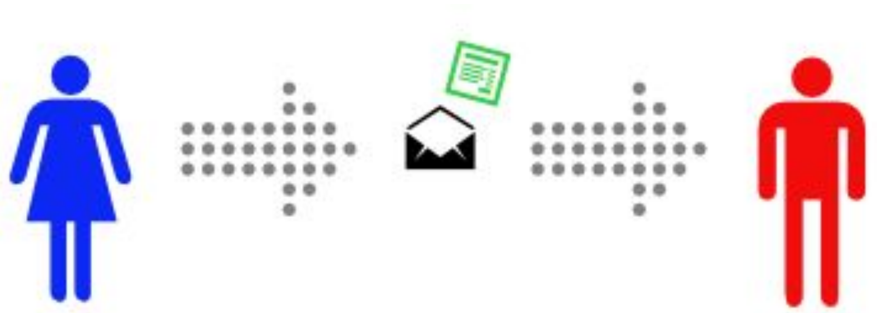
Project 2

Leo Ho, Matteo Mastrogiovanni, Nathan Cantu

Project Type

Project Type 1

Party 1 emails/sends a message to Party 2



Possible Attacks

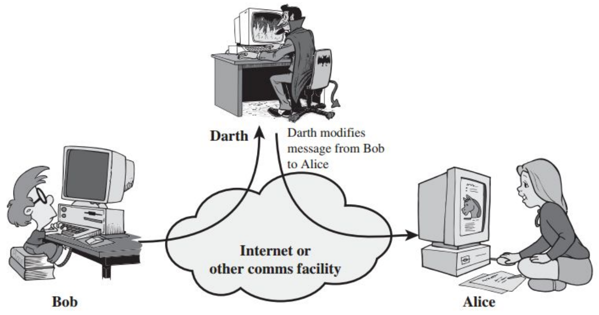
- Data eavesdropping
- Data Modification
- Data Originator Spoofing
- Data Replay (MITM)

Data Eavesdropping

- Also known as sniffing or snooping attacks
- Attacker reroutes network traffic and passively monitors data that is transmitted
- Any unencrypted data transmitted over an open network is susceptible to this attack
- Examples: WiFi hotspots, websites without HTTPS
- Types of attacks: MITM, Skype & Type, Keystroke Logging

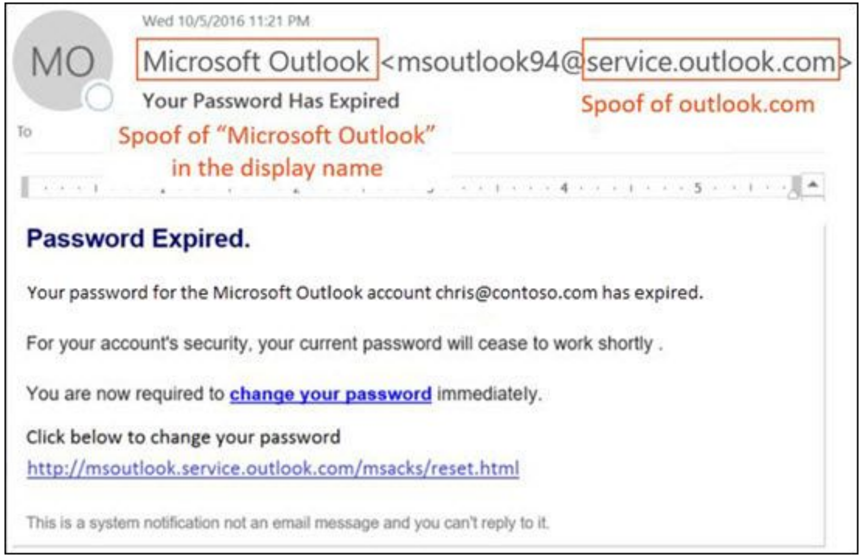
Data Modification

- Attacker changes, deletes, or adds contents to a message between two parties
- Active attack: Attacker has access to continuous stream of messages
- Example: and email stating “Allow Bob to read confidential item X” to “Allow Alice to read confidential item X”


Darth
Darth modifies message from Bob to Alice
Internet or other comms facility
Bob
Alice

Data Originator Spoofing

- Email Spoofing: email messages with forged sender addresses
- Includes phishing emails and spam
- Attacker attempts to trick user to relay sensitive information or download malware


Wed 10/5/2016 11:21 PM

MO Microsoft Outlook <msoutlook94@service.outlook.com>

Your Password Has Expired

To

Spoof of "Microsoft Outlook" in the display name

Spoof of outlook.com

Password Expired.

Your password for the Microsoft Outlook account chris@contoso.com has expired.

For your account's security, your current password will cease to work shortly .

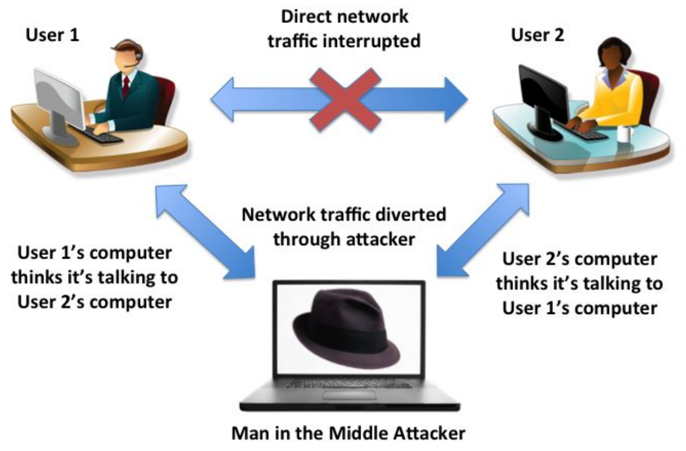
You are now required to **change your password** immediately.

Click below to change your password
http://msoutlook.service.outlook.com/msacks/reset.html

This is a system notification not an email message and you can't reply to it.

Man-in-the-middle Attacks

- Attacker gains access to server that email or messages are being sent through
- Can pose as both users while stealing any information that is passed between the parties
- For email, this is most likely accomplished through IP Spoofing or WiFi eavesdropping


User 1
Direct network traffic interrupted
User 2
Network traffic diverted through attacker
User 1's computer thinks it's talking to User 2's computer
User 2's computer thinks it's talking to User 1's computer
Man in the Middle Attacker

Design Methods

Design methods

- AES-GCM for encryption
- Diffie-Hellman for key exchange

Design Security

- Diffie-Hellman allows for the exchange of public keys where users combine keys to encrypt and decrypt messages
- Secure against eavesdropping but is susceptible to MITM attacks and authentication without additional encryption
- AES-GCM encrypts the data being sent between two parties using the shared private key provided by Diffie-Hellman
- AES-GCM is efficient (stream cipher) and allows for authentication, two important aspects of email security
- Since only public keys are shared and used in encryption, this protects against spoofing and data modification

AES-GCM

AES-GCM (Galois/Counter Mode)

- GCM is a mode of operation for symmetric-key cryptographic block ciphers which is widely adopted for its performance
- The operation is an authenticated encryption algorithm designed to provide both data authenticity (integrity) and confidentiality
- GCM is defined for block ciphers with a block size of 128 bits

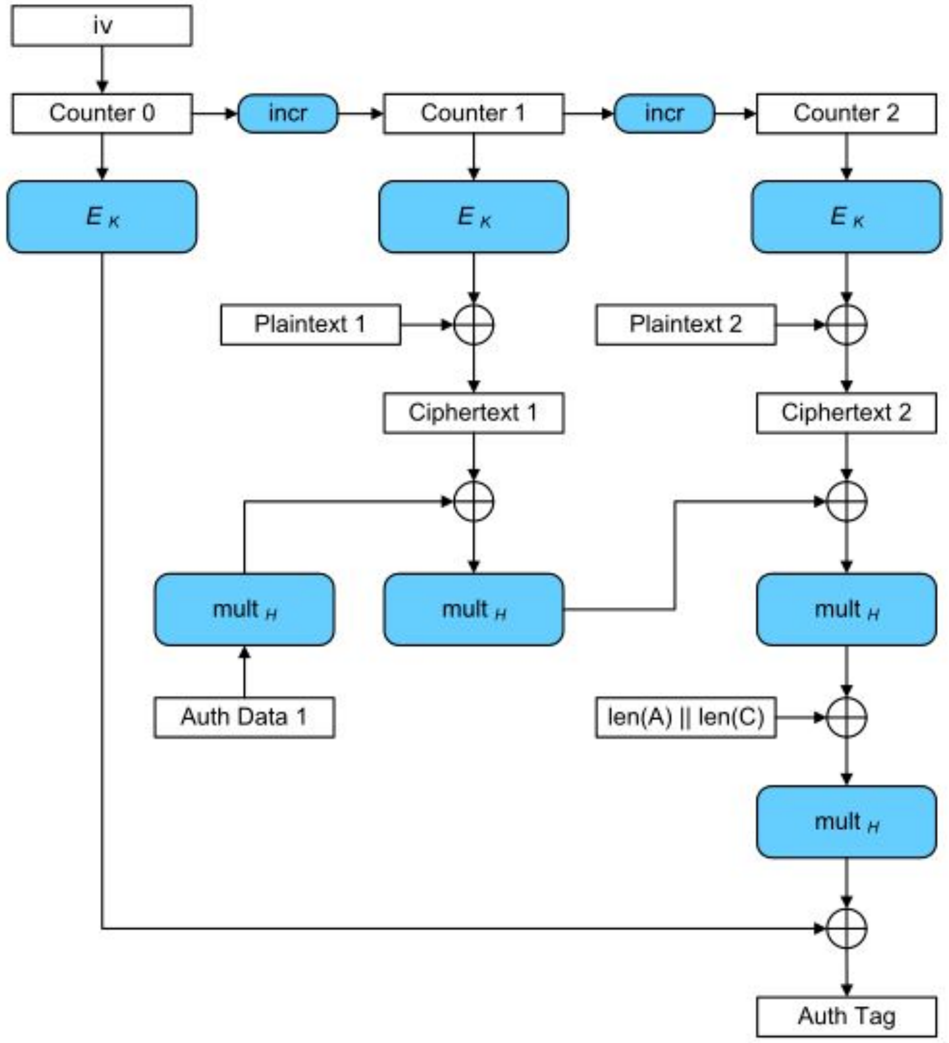
AES-GCM’s Basic Operation

- Similar to normal counter mode, blocks are numbered sequentially
- The block number is combined with an initialization vector (IV) and encrypted with a block cipher E, usually AES
- The result is XORed with the plaintext to obtain the ciphertext
- This is a stream cipher, so different IV must be used for each stream that is encrypted

AES-GCM’s Basic Operation

- The ciphertext blocks are considered coefficients of a polynomial which is evaluated at a key-dependent point H, using finite field arithmetic
- The result is encrypted to produce an authentication tag used to verify the integrity of the data
- The encrypted text contains the IV, ciphertext, and authentication tag

AES-GCM's basic operation



AES-GCM’s Mathematical Basis

- The key-feature is the ease of parallel-computation of the Galois field multiplication used for authentication
- The GF(2^128) field is defined by the polynomial:

$$x^{128} + x^7 + x^2 + x + 1$$

AES-GCM’s Mathematical Basis

- The authentication tag is constructed by feeding blocks of data into the GHASH function

  GHASH(H,A,C) = X_{m+n+1}

  where H = E_k(0^128) is the Hash key, a string of 128 zero bits encrypted using the block cipher, A is the data which is only authenticated (not encrypted), and C is the ciphertext, m is the number of 128-bit blocks in A, n is the number of 128-bit blocks in C (both round up)

AES-GCM’s Mathematical Basis

- The authenticated text and ciphertext are separately zero-padded to multiples of 128 bits and combined into a single message

$$S_i = \begin{cases} A_i & \text{for } i = 1, \ldots, m-1 \\ A_m^* \parallel 0^{128-v} & \text{for } i = m \\ C_{i-m} & \text{for } i = m+1, \ldots, m+n-1 \\ C_n^* \parallel 0^{128-u} & \text{for } i = m+n \\ \text{len}(A) \parallel \text{len}(C) & \text{for } i = m+n+1 \end{cases}$$

where len(A) and len(C) are 64-bit representations of the bit lengths of A and C, v = len(A) mod 128, u = len(C) mod 128, and

|| is concatenation

AES-GCM’s Mathematical Basis

- X_i is defined as:

$$X_i = \sum_{j=1}^{i} S_j \cdot H^{i-j+1} = \begin{cases} 0 & \text{for } i = 0 \\ (X_{i-1} \oplus S_i) \cdot H & \text{for } i = 1, \ldots, m + n + 1 \end{cases}$$

AES-GCM's Performance

- GCM requires one block cipher operation and one 128-bit multiplication in the Galois field per each block of encrypted and authenticated data
- The block cipher operations and multiplication operations can be pipelined or parallelized
- Some impressive performance results are published on a number of platforms
- Shay Gueron and Vlad Krasnov achieved 2.47 cycles per byte on the 3rd generation Intel processors

AES-GCM’s Security

- It is secure when it is used with a block cipher that is indistinguishable from a random permutation
- However, security depends on choosing a unique initialization vector for every encryption performed with the same key
- For any given key and initialization vector combination, GCM is limited to encrypting 64 GiB of plaintext.
- The authentication strength depends on the length of the authentication tag

Diffie–Hellman

Diffie–Hellman

- Method for securely exchanging cryptographic keys over a public channel (between the sender and receiver sockets)
- One of the first public-key protocols

- Goal: Using a common encryption algorithm between two parties, construct a common encryption key that defends against eavesdropping attacks

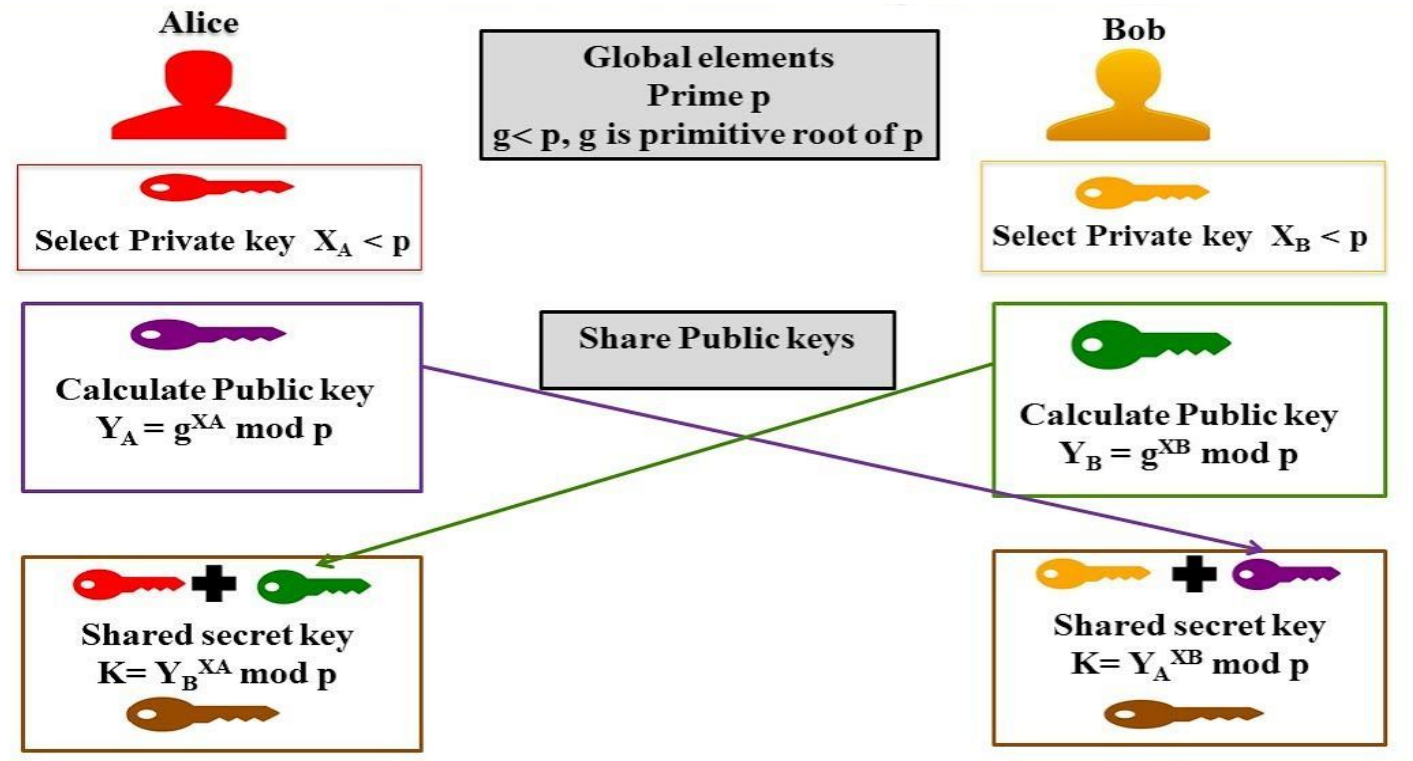
Diffie-Hellman Basic Operations

- Both publicly agree on a prime modulus and a generator (p and g respectively)
- Private keys are selected for both the sending and receiving sockets, in this implementation the keys are integers chosen randomly within the range of 1 to p-1
- Using their specific private key value and the public p and g values, both sockets calculate and send a resultant public key to the other socket

Diffie-Hellman Basic Operations

- Both sockets then take the other’s public key and raise it to the power of their private key
- The value is modded by the prime modulus p
- The result is the shared common encryption key
- Both socket calculations result in the same private key, thus it does not need to be transmitted over the network, allowing it to stay completely private (the attacker does not know the encryption key)

Diffie-Hellman Basic Operations



Diffie–Hellman Mathematical Basis

- G is a primitive root of value p and is often a small prime number
- P has to be a large prime for adequate security yet has to be efficient
- P is usually between 2000-4000 bits (for this implementation the length was 4096 bits)
- Based on the modulo operation % (1 to p-1) and exponentials

Diffie-Hellman Security

- Though the private key is within the range of 1 to p-1, p is always going to be large enough in that a brute force attack is infeasible
- Because you are taking the mod of an extremely large number (g^a), the attacker cannot determine what the private key is knowing g, p, and the resulting public key
- Every number greater than p when modded will begin at 1 again, thus if the attacker knows the private key and p and g, the private key would be a relatively infinite number of possibilities making it virtually impossible to determine within a feasible amount of time

Diffie–Hellman Security

- If a represents the private key of Alice and b represents the private key of bob, in order to get the resulting public keys, the calculations would be g^a mod p or g^b mod p
- In the second round, Bob takes the public key sent by alice and raises it to the power of his private key, thus = (g^a)^b mod p
- Alice does the same = (g^b)^a mod p
- Since (g^b)^a = (g^a)^b = g^ab then the answer for both will be the same (g^ab) mod p
- Both produced the same encryption key without knowing what the other’s private key was!

Diffie–Hellman Security

- At most the attacker can know g^a+b not g^ab since g^a and g^b are made public
- Does not protect against man in the middle attacks unless paired with additional satisfactory encryption
- It does not provide authentication thus the adversary could pose as the recipient and receive information that allows them to decrypt and re-encrypt the messages
- AES-GCM provides this additional protection

Demo