Security of Symmetric Encryption against Mass Surveillance

Gihyuk Ko

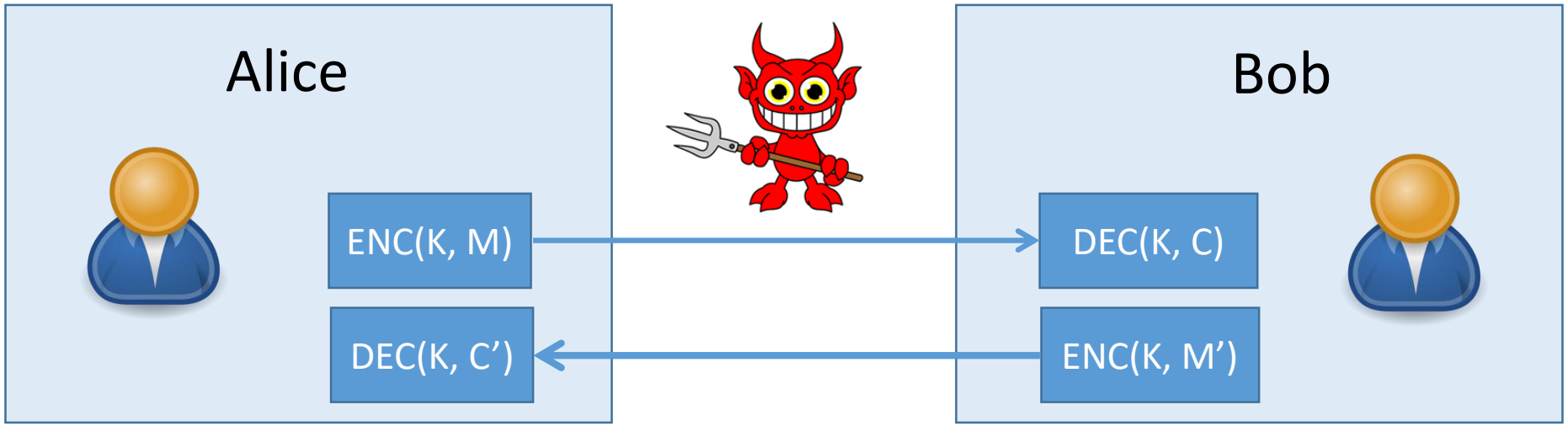
Carnegie Mellon University

slides from Kyle Soska

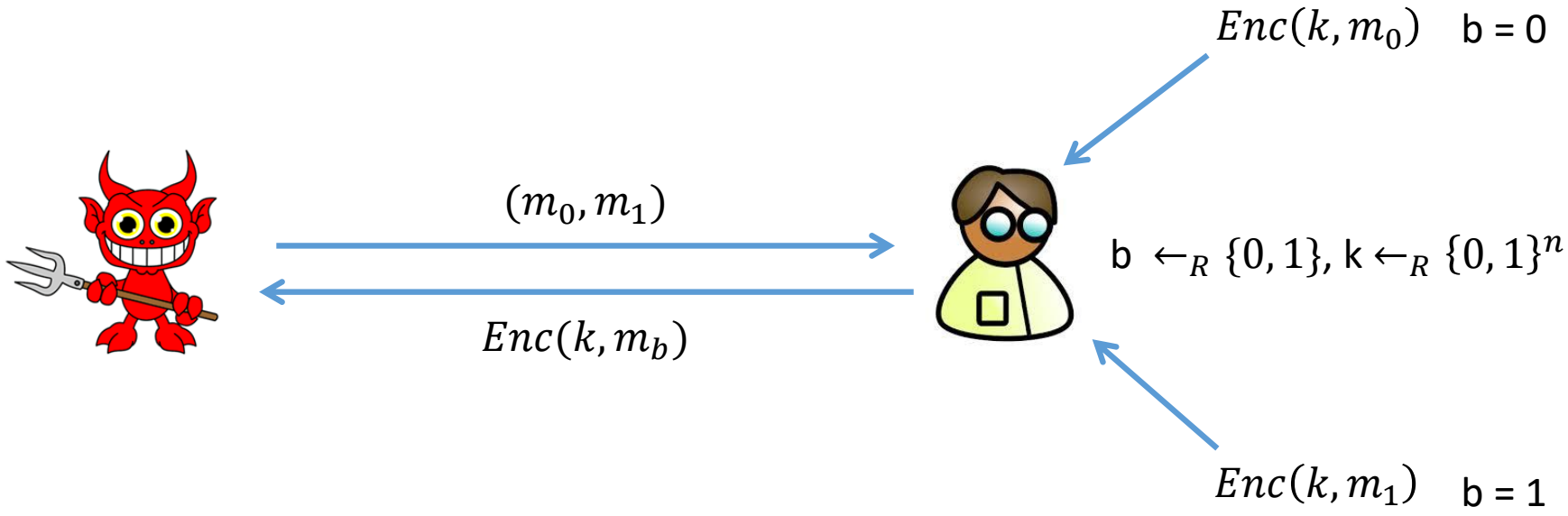
Today’s Lecture

- Consider new type of adversary in the form of a ‘Big Brother’ that wants to perform surveillance at a large scale
- Discuss constructions that could be used by a ‘Big Brother’ for symmetric encryption
- Discuss mitigations and solutions against these constructions

Traditional Adversary Model



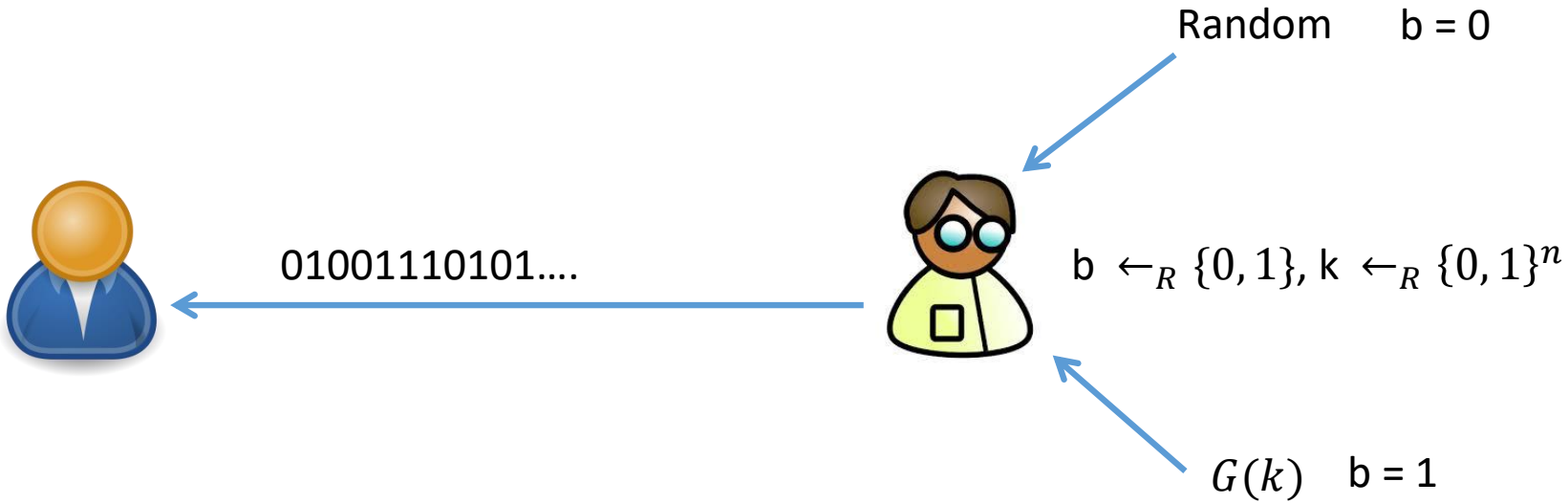
Example: IND-CPA



$$Adv(A) = \Pr\left[A\left(Enc(k, m_0)\right) = 1\right] - \Pr\left[A\left(Enc(k, m_1)\right) = 1\right]$$

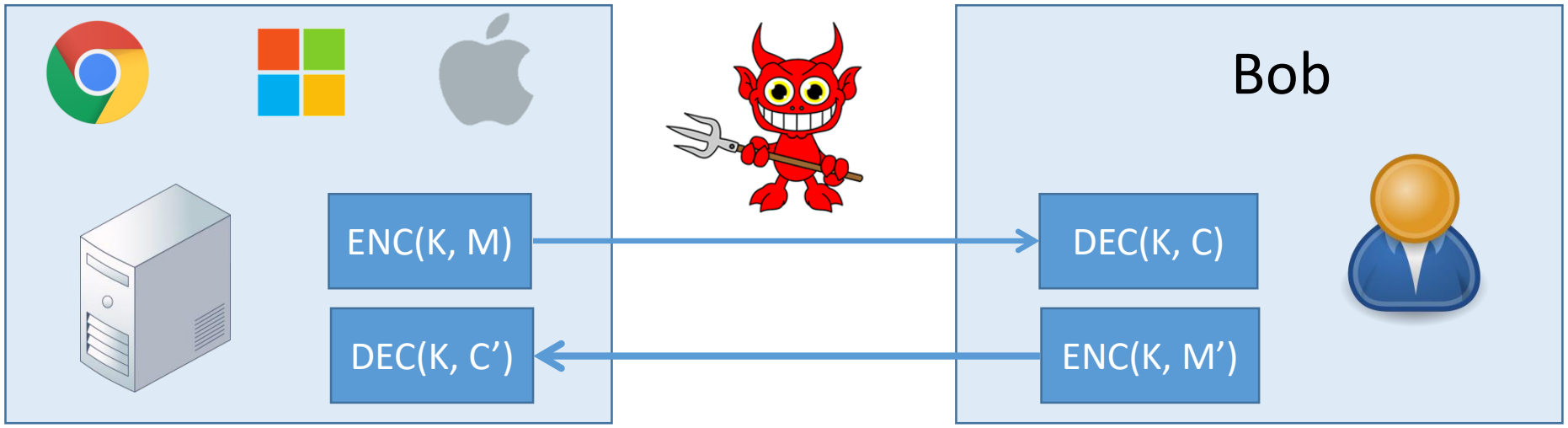
Example: Secure PRNG

- Ideal: Sequence of truly random bits
- Actual: Pseudorandom sequence of bits

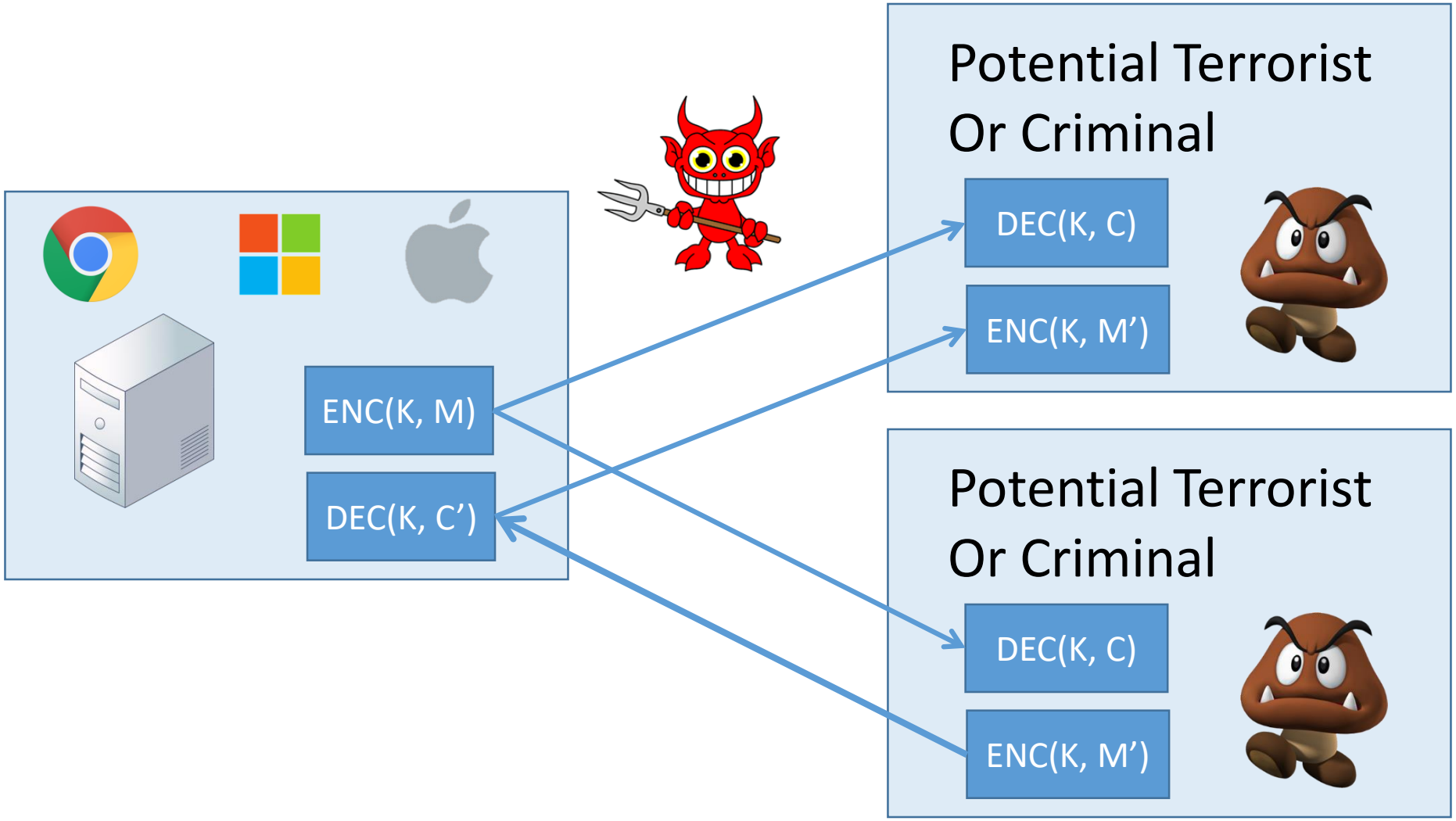


$$Adv(A) = \Pr[A(G(k)) = 1] - \Pr[A(r) = 1]$$

Traditional Adversary Model



Terrorism Adversary Model



Terrorism and Large Internet Companies

Twitter's New ISIS Policy

CNN U.S. | World | Politics | Money | Opinion | Health | Entertainment | Style | Travel | Sports | Video Live TV U.S. Edi

Time social media

By Ted

Upda

"f

The long read

How the changing media is changing terrorism

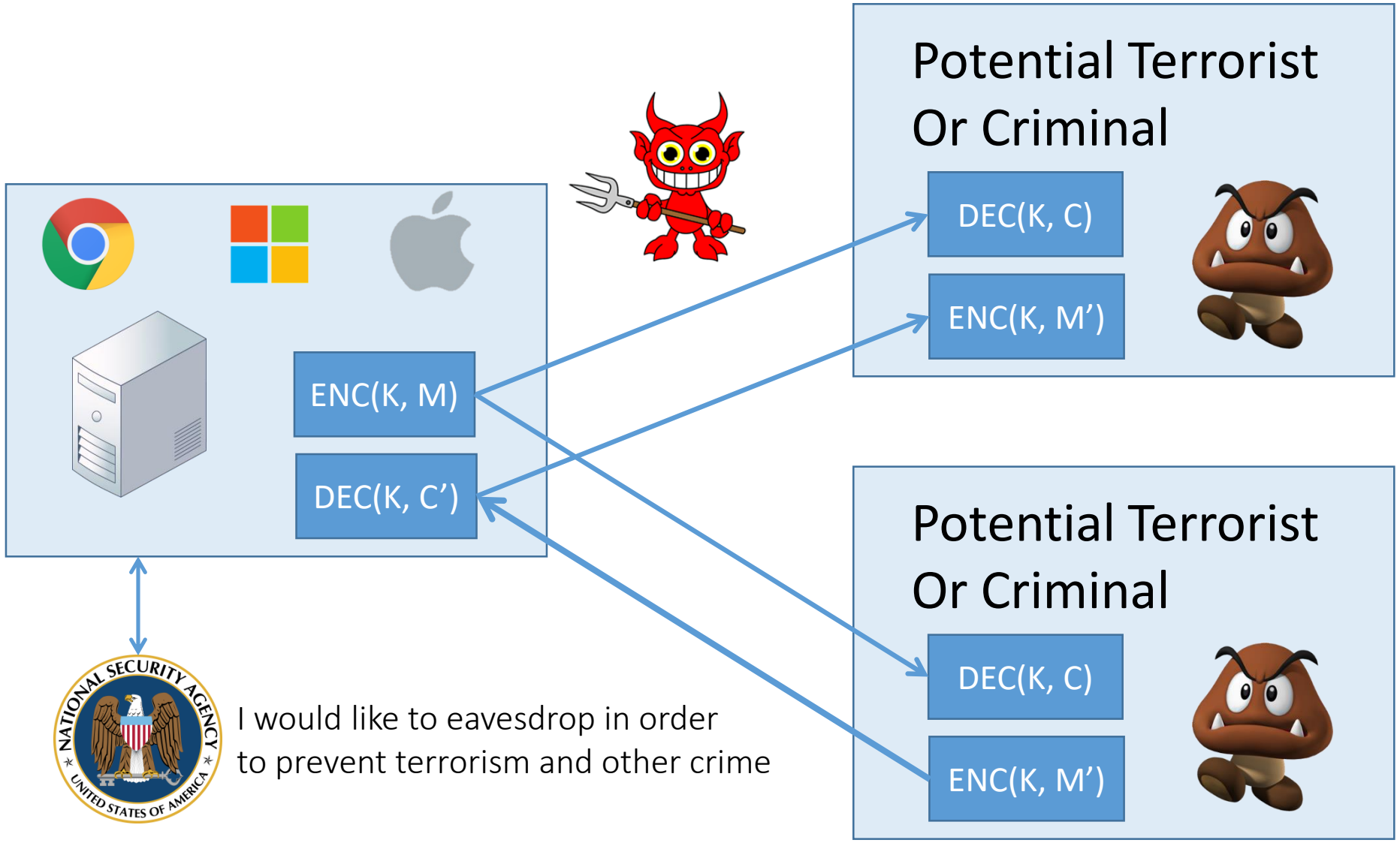
Just like news organisations, terrorists need an audience - and both have adapted their tactics to keep your attention

by Jason Burke

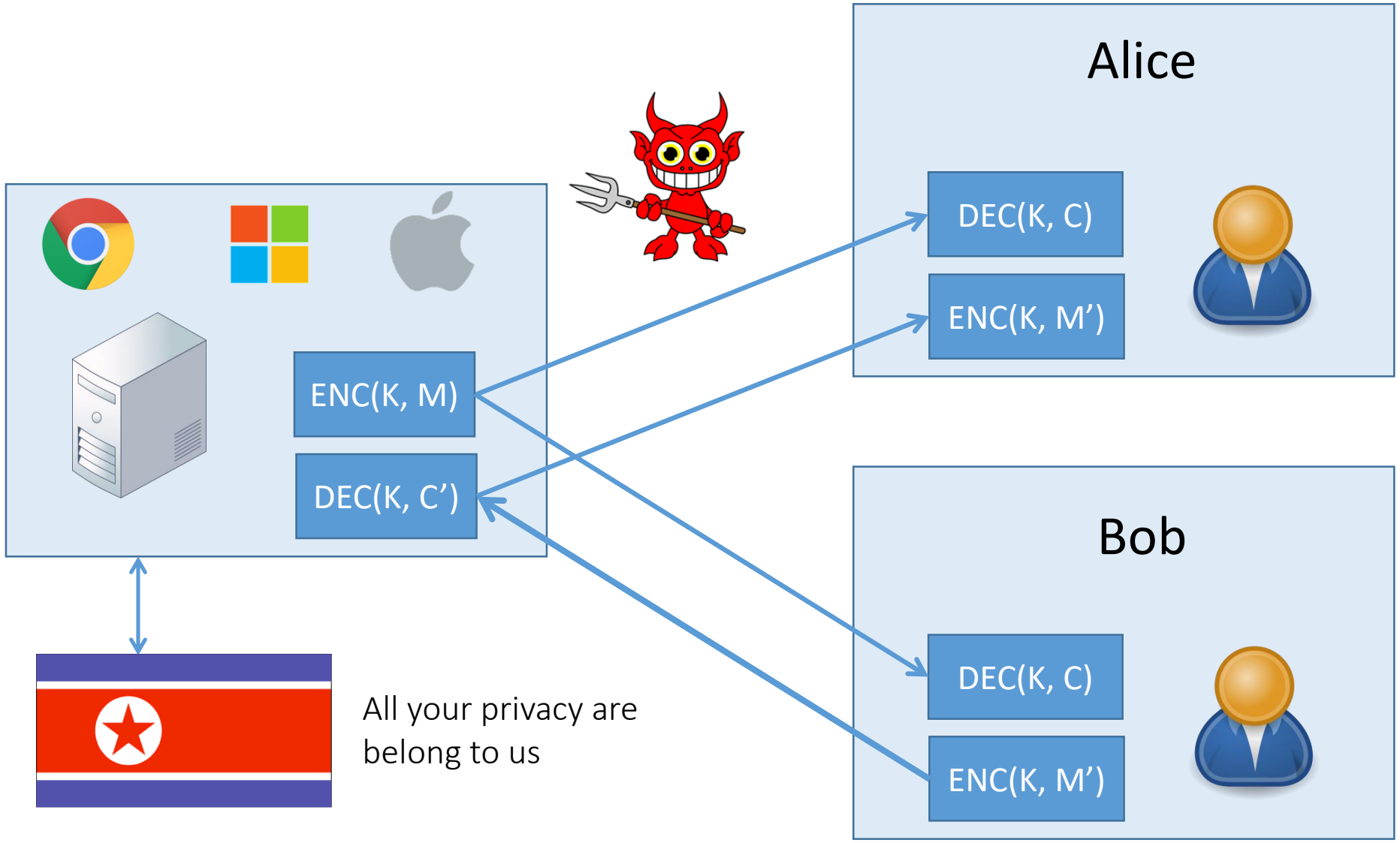
Federal Insider

Homeland Security to amp up social media screening to stop terrorism, Johnson says

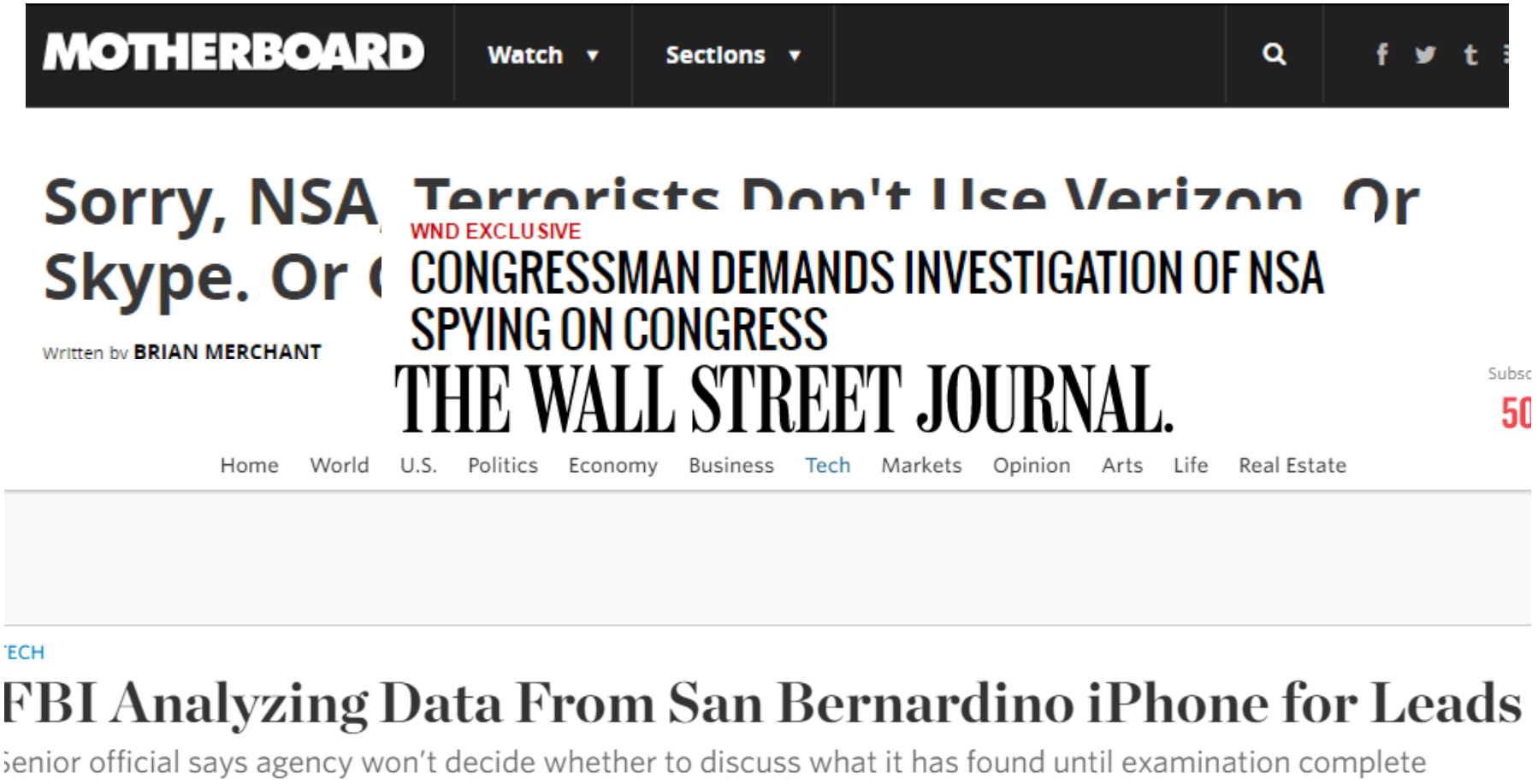
Terrorism Adversary Model



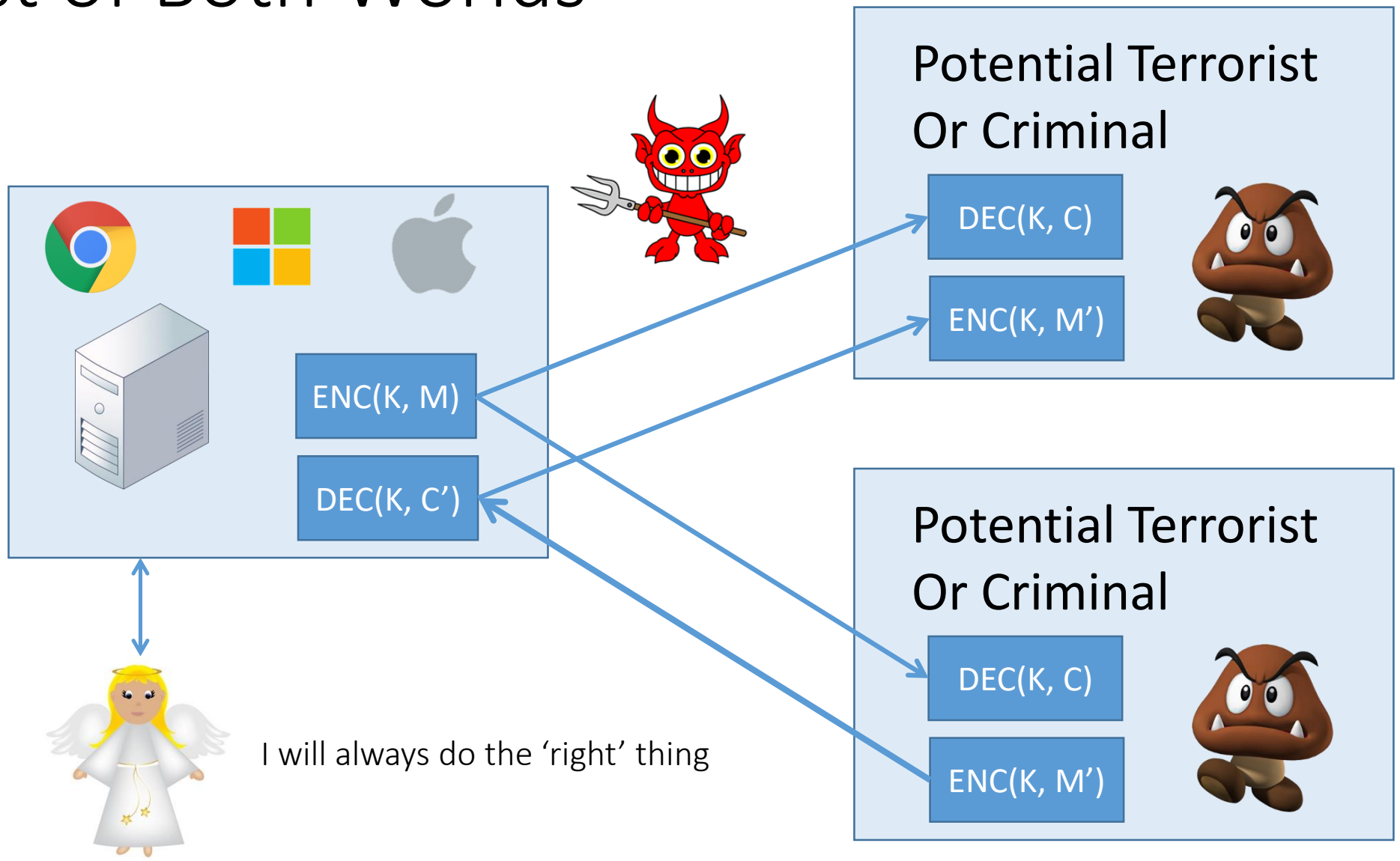
Spying Government Adversary Model



Spying Government



Best of Both Worlds



The Rest of the Lecture

- How could a company allow a third party such as the government to eavesdrop in an undetectable way?

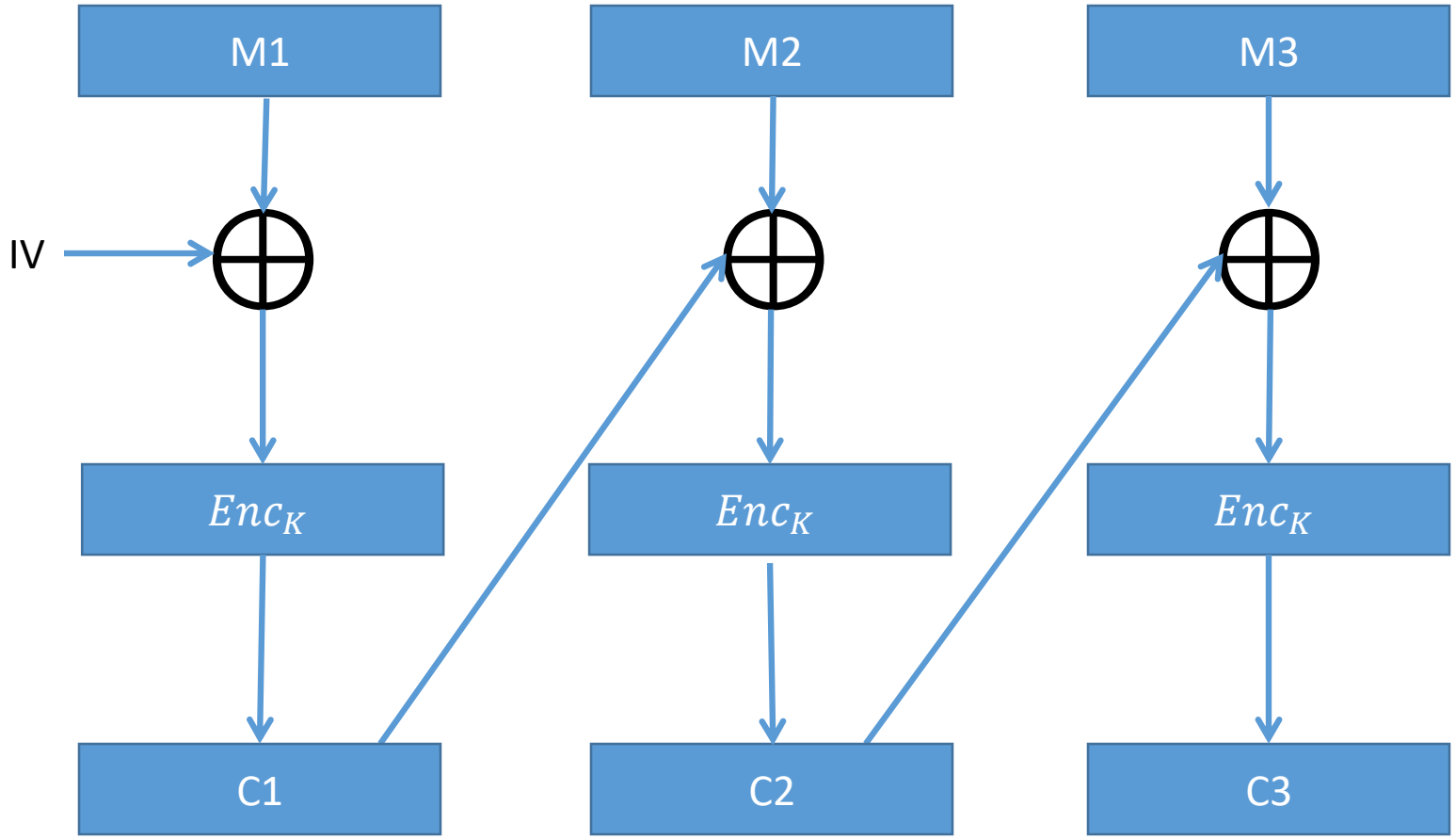
- What can we as users do to prevent such things from happening?

Goals - Informal

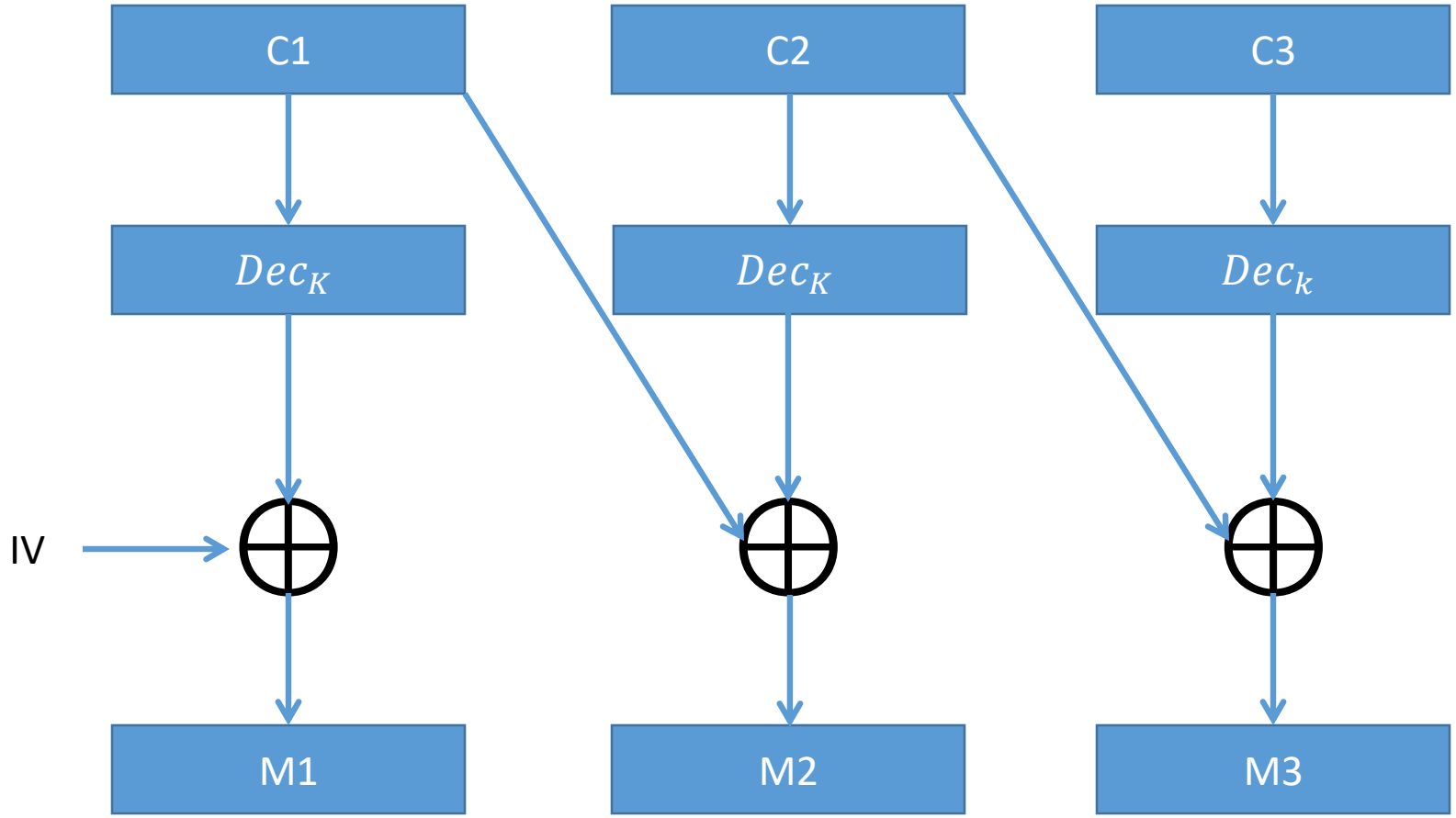
- **Big Brother ($\boldsymbol{B}$)**
  - Wants to eavesdrop on communication
  - Does not want its eavesdropping to be detected

- **Users ($\boldsymbol{U}$)**
  - Wants to detect when eavesdropping is taking place
    - Want to prevent eavesdropping from taking place

Recall CBC Encryption



Recall CBC Decryption



Encryption Schemes More Formally

- $\Pi = (K, E, D)$
  - $\mathcal{K} = \{0,1\}^k$ the keyspace
  - K: Secret symmetric key
  - M: Message
  - A: Associated Data (Supplementary information such as padding)
  - σ: state (ex. A counter for CBC counter mode)

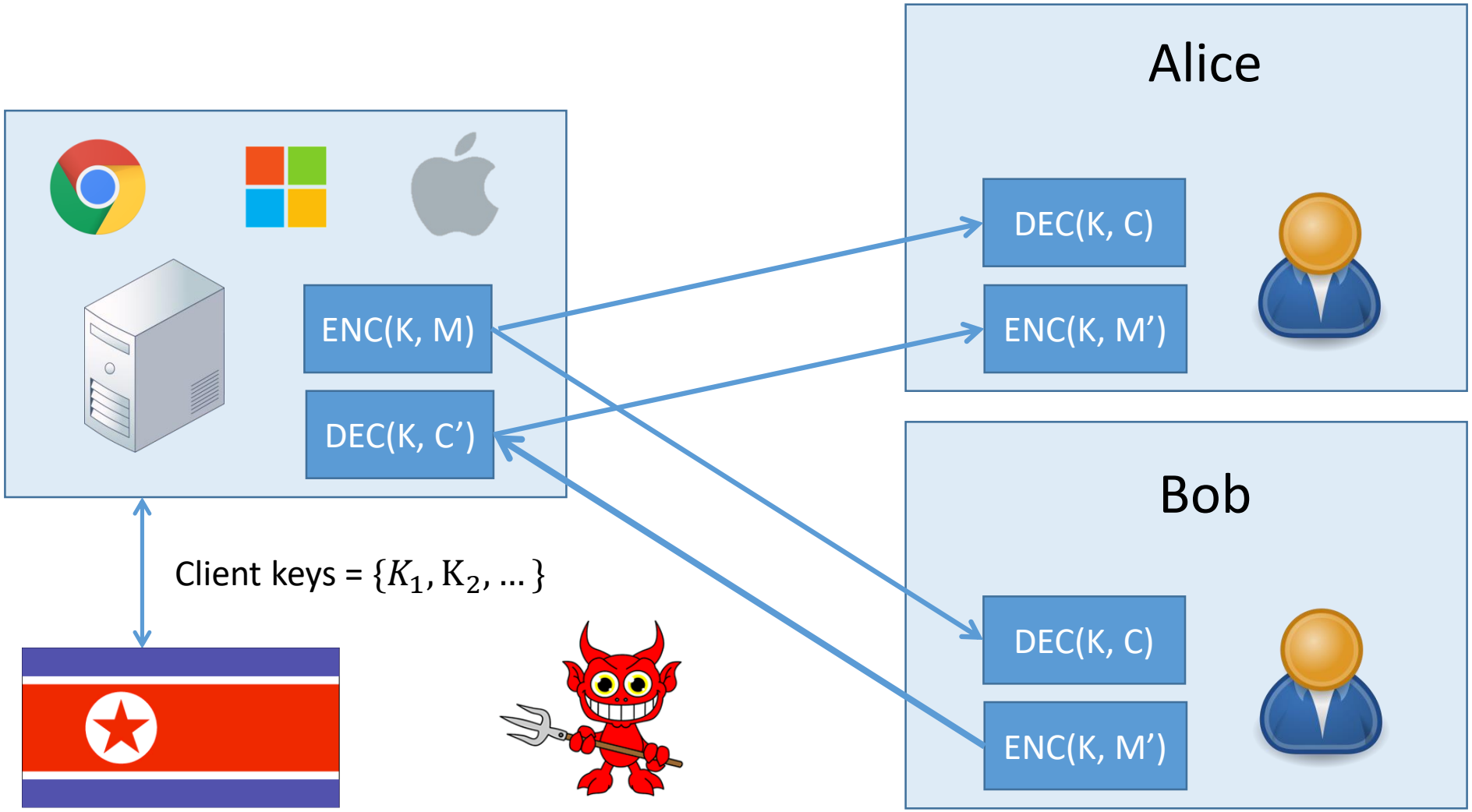
  - E is a possibly randomized encryption routine such that $(C, \sigma') \leftarrow E(K, M, A, \sigma)$

  - D is a deterministic decryption routine such that $(M, \sigma') \leftarrow D(K, C, A, \sigma)$

How to subvert Π?

- Transfer all symmetric keys K to B
  - Keys would be observed by an eavesdropper and surveillance would be detected
  - If the company ever wanted to stop sending keys to B, then B would no longer be able to eavesdrop

Spying Government Adversary Model

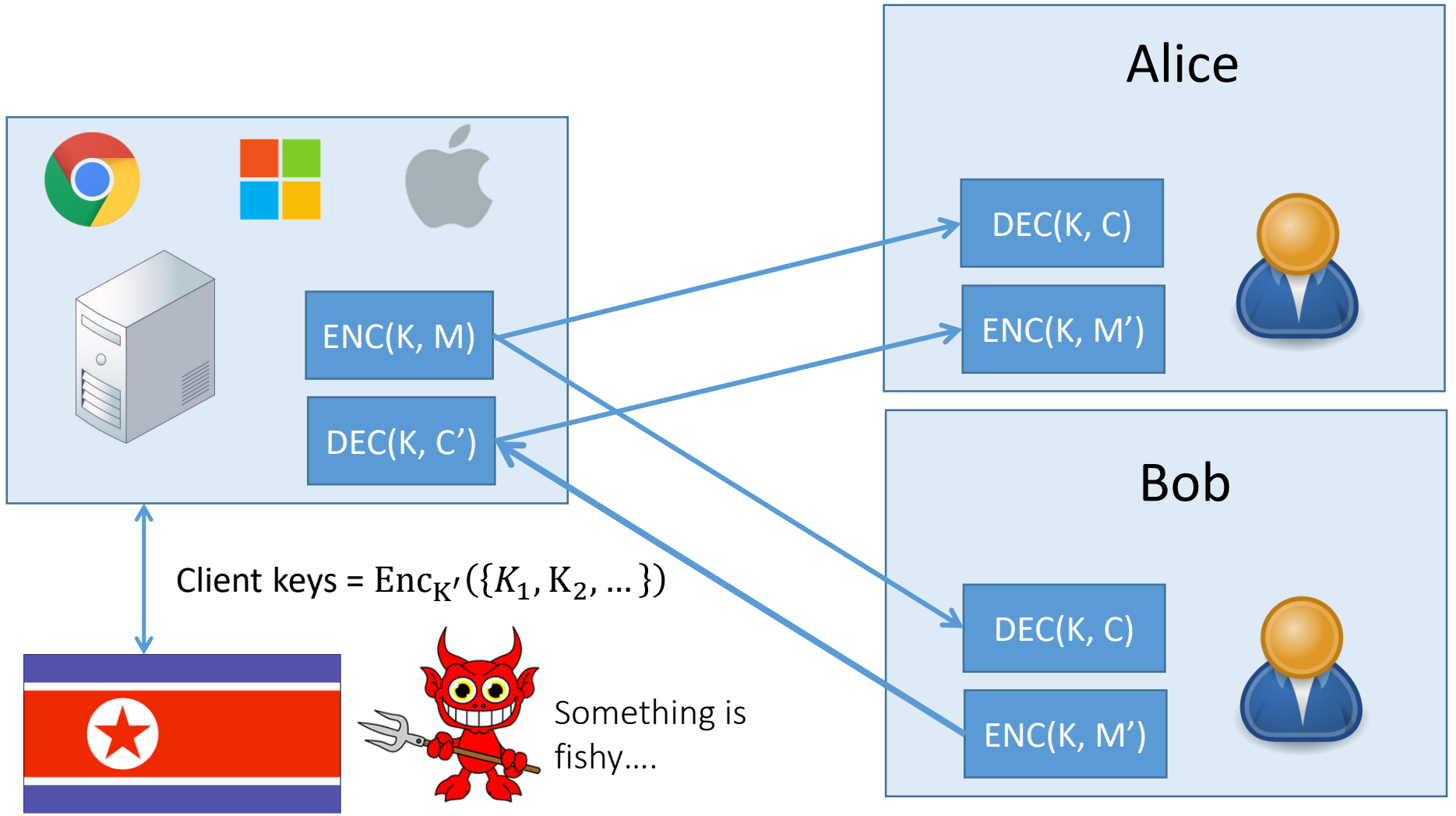


How to backdoor Π?

- Transfer all symmetric keys K to B
  - Keys would be observed by an eavesdropper and surveillance would be detected

- Transfer all symmetric keys encrypted under key K'
  - Still high bandwidth communication channel between company and surveillance body, suspicious

Spying Government Adversary Model



Algorithm Substitution Attack (ASA)

- Suppose the servers replaced their implementation of encryption scheme Π with a modified version Π'
- Π' could be specially designed to leak information about the messages or secret keys to an eavesdropper who holds secret information K'
  - Eavesdroppers that do not know this secret information would not be able to learn the messages or K
- The server's implementation of Π is a black box from the point of view of a regular user
- Π' has all the information Π has as well as a key K' called the escrow key or big-brother key, and possibly more state

How to backdoor Π?

- Derive a modified Π' from Π, what properties must Π' have?

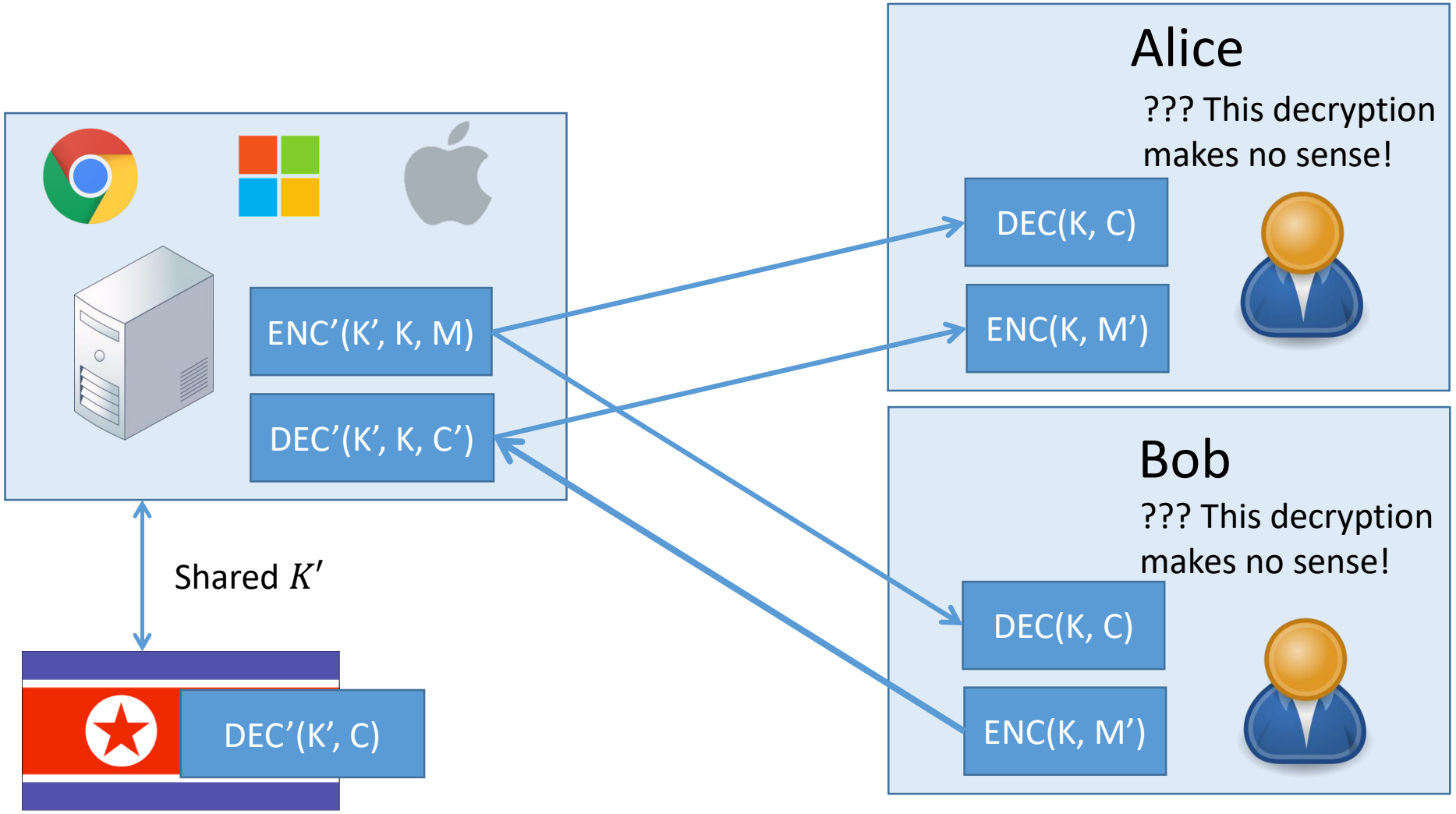
Decryptability (Informal)

- The modified $\Pi' = (K', E', D')$ must produce encryptions that are correctly decrypted by the unmodified $\Pi = (K, E, D)$

Decryptability (Formal)

- $\Pi' = (K', E', D')$ satisfies decryptability relative to $\Pi = (K, E, D)$ if (K', K, E', D') is a correct encryption scheme where D' is defined by $D'\big((K', K), C, A, \sigma\big) = D(K, C, A, \sigma)$

If decryptability does not hold



How to backdoor Π?

- Derive a modified Π' from Π, what properties must Π' have?

Decryptability (Informal)

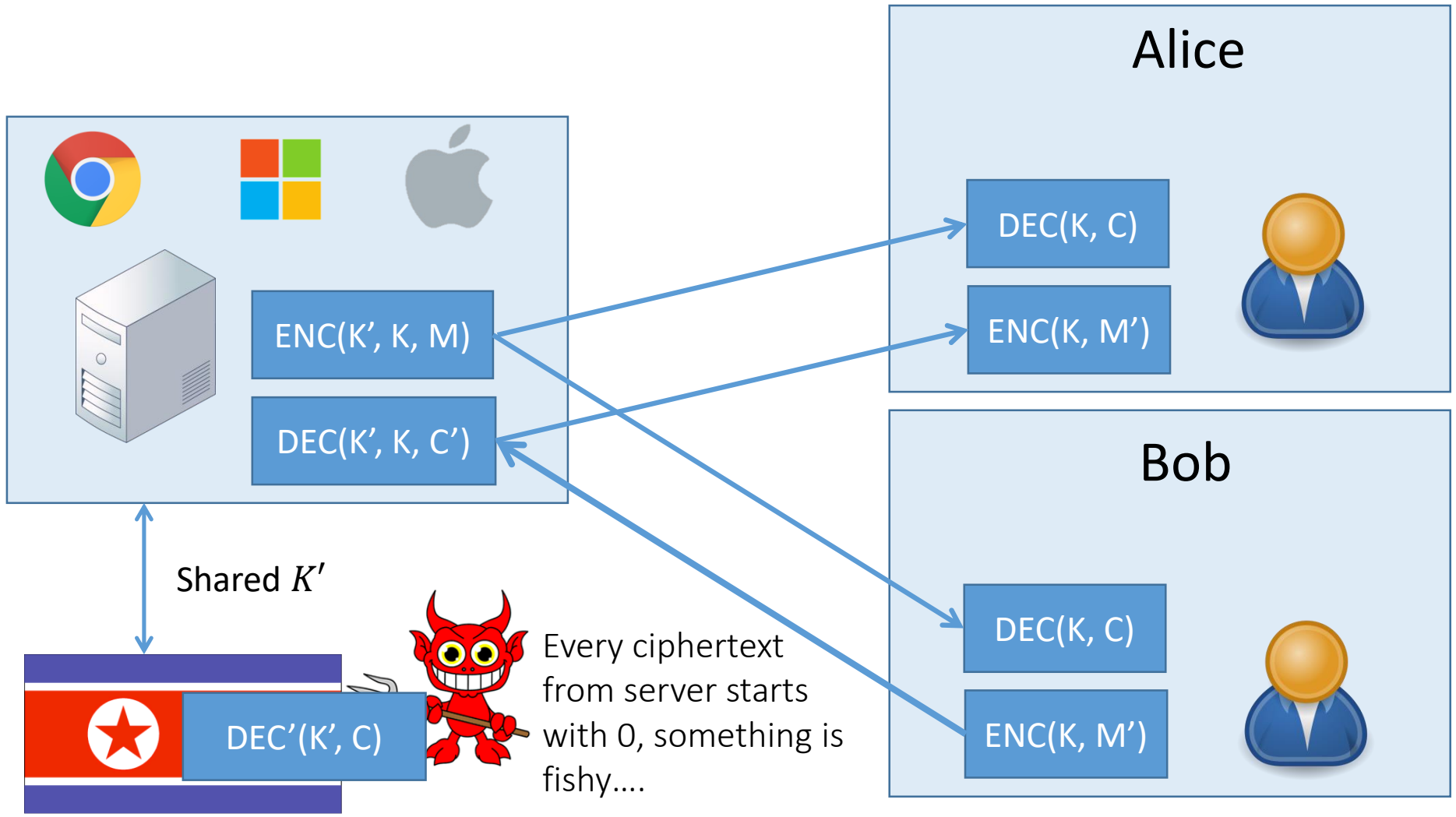
- The modified $\Pi' = (K', E', D')$ must produce encryptions that are correctly decrypted by the unmodified $\Pi = (K, E, D)$

Decryptability (Formal)

- $\Pi' = (K', E', D')$ satisfies decryptability relative to $\Pi = (K, E, D)$ if (K', K, E', D') is a correct encryption scheme where D' is defined by $D'\big((K', K), C, A, \sigma\big) = D(K, C, A, \sigma)$

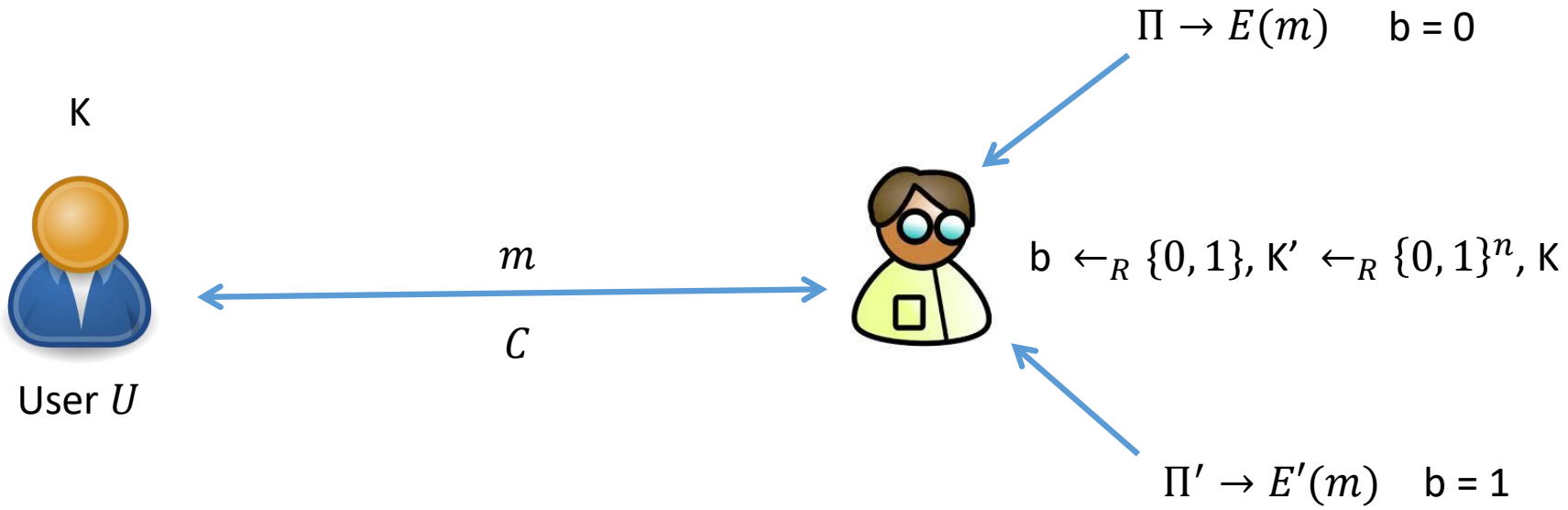
- Needs to 'look like' a regular ciphertext, can't have anything abnormal such as ciphertexts always start with a '0'

If ciphertexts are distinguishable



Detection Game

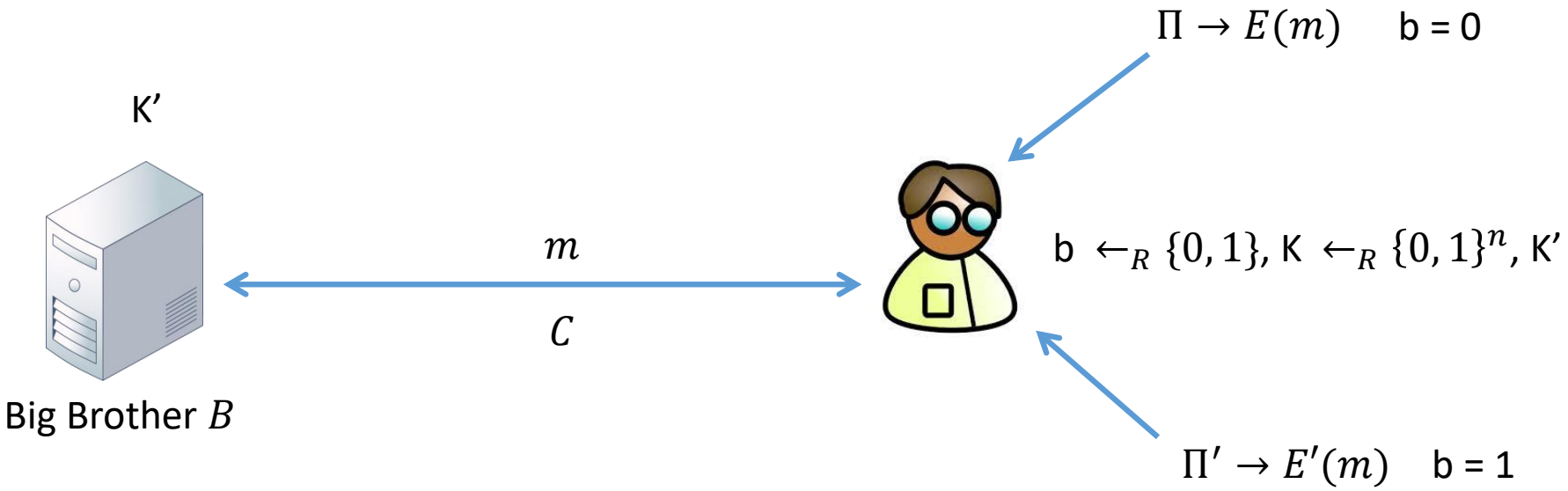
- Can a user with secret key K tell the difference between encryptions with E and E'?



$$Adv(U) = \Pr\left[U\left(K, E'(m)\right) = 1\right] - \Pr\left[U\left(K, E(m)\right) = 1\right]$$

Surveillance Game

- Can Big brother with escrow key K' tell the difference between E and E'?



$$Adv(B) = \Pr\left[B\left(K', E'(m)\right) = 1\right] - \Pr\left[B\left(K', E(m)\right) = 1\right]$$

Additional Comments

- Practically, Big Brother would want to be able to do more than simply distinguish encryptions from E and E' with non-negligible probability
- We will be looking at schemes that allow Big Brother to either completely learn each message, or learn the session key so that he can decrypt each message with advantage nearly 1
- Big brother wants to win surveillance game with advantage 1 and for users to win detection game with negligible probability
- Users want to win detection game with advantage 1, and for big brother to win surveillance game with negligible probability

IV Replacement Attacks

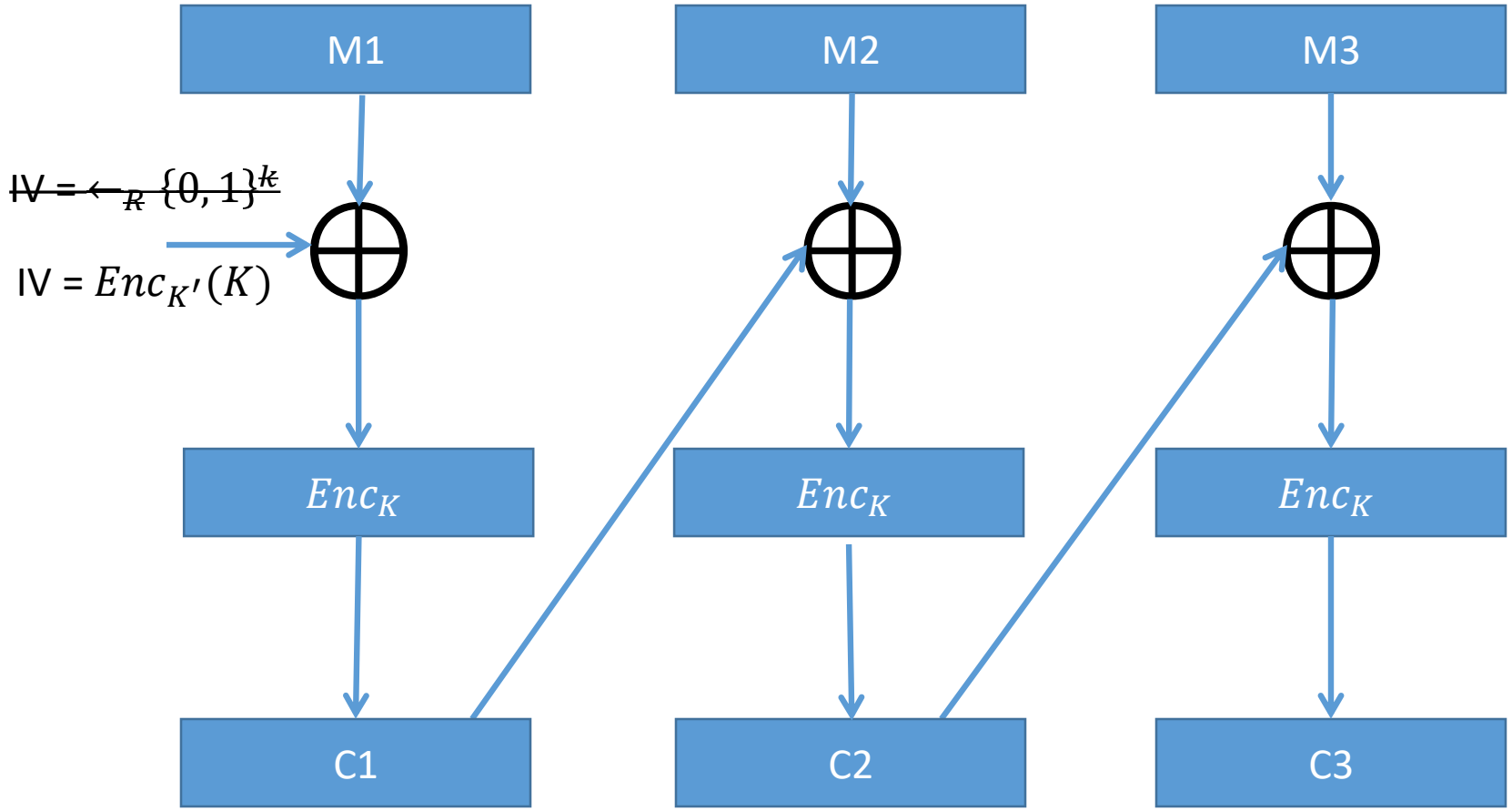
- Lots of encryption schemes flip coins (generate random information)
  - Turing machine with a random tape

- Idea: when Π' needs to generate random information, it will instead select the values specifically for the purpose of leaking information
  - Nonces
  - Initialization Vectors (IV)
  - Variable Length Padding

  - Can a user detect this type of attack?

- Let X be some efficient algorithm that extracts $IV \leftarrow X(C)$ from ciphertext C, if such an algorithm exists then the IV is said to be 'surfaced' or the encryption scheme 'surfaces' its IV

Stateful IV Attack



An Example Problem – part (a)

- Question

Suppose for all messages m and keys k, |E(m,k)| = |m|, that is, the size of the encryption of m (in bits) is the same as the size of m. Argue that (E,D) cannot be CPA-secure.

- Answer
  - Domain and Range of the encryption is the same: it has to be deterministic!
  - Deterministic encryption cannot be secure against CPA adversary!

Fixed IV attack

Question

- Can we just set a fixed IV = $Enc_{K'}(K)$ for every message? What is the problem with this scheme?

Answer

- IV will always be the same, this is easily detectable by a user!

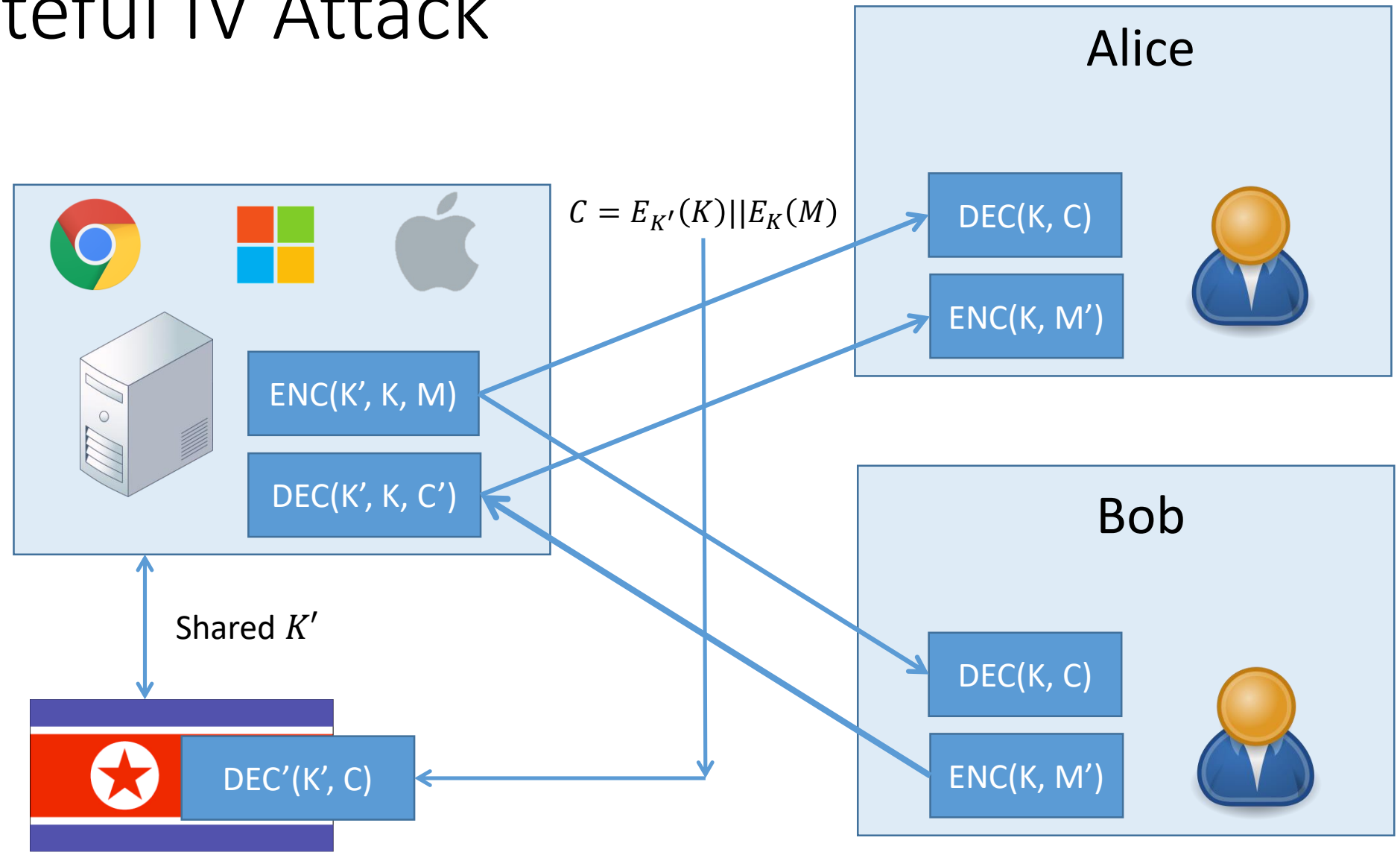
- How can we modify this to make the attack work?

Stateful IV Attack

- $\boldsymbol{E'(K', \mathbf{K}, \mathbf{M}, \mathbf{A}, \sigma)}$ Algorithm:
  - Let σ be a counter initialized to 0
  - If $\sigma = 0$ then $IV \leftarrow Enc_{K'}(K)$
  - Else $IV \leftarrow_R \{0,1\}^k$
  - $C = E_K(M, A, IV)$
  - $\sigma = \sigma + 1$; Return C

- $\boldsymbol{D'(K', C, A)}$ Algorithm ($\boldsymbol{C}$ is an indexed array of ciphertexts):
  - $IV \leftarrow X(C[1])$
  - $K \leftarrow Dec_{K'}(IV)$
  - $M[1] \leftarrow D_K(C[1], A[1])$
  - Return M

Stateful IV Attack



Attack Against Stateful IV Attack

- State reset attack
  - If the state could get reset to $\sigma = 0$ then the IV would repeat, highly unlikely to happen if the IV is truly random or pseudorandom

- Solution
  - Use a probabilistic / combinatorial version of Π instead of a stateful version of Π

Stateless IV Attack

- The intuition is that we will randomly select a single bit of the Key K and leak it in an IV

- $IV = Enc_{K'}(bit, index, random\ pad)$

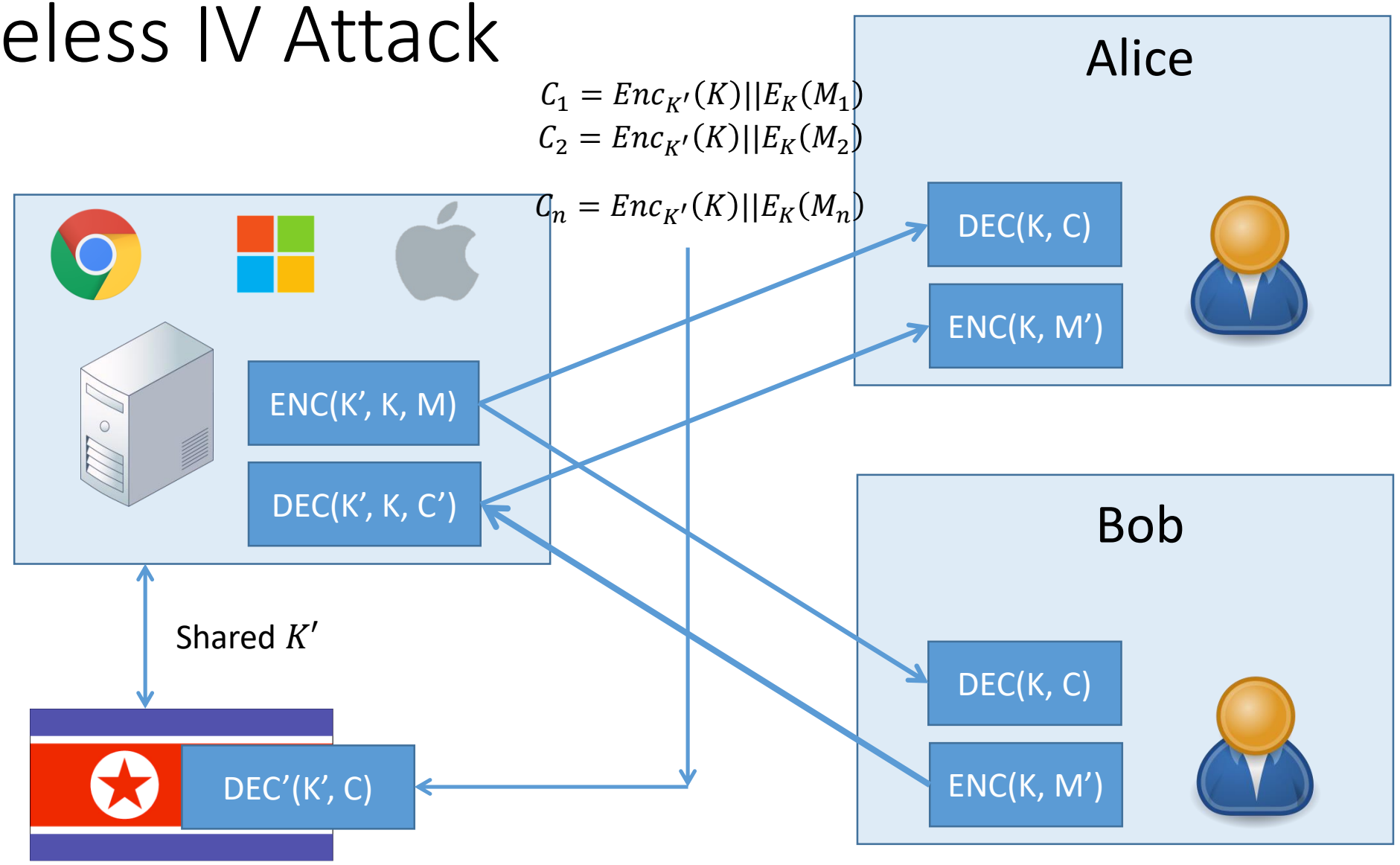
Stateless IV Attack

- Let k be the size of the key (ex. 128 bits), and let $v = \log_2 k$
- Notation: $K[i]$ refers to a single bit of K at index i

- $E'(\mathrm{K}', \mathrm{K}, \mathrm{M}, \mathrm{A}, \sigma)$ Algorithm:
  - $l \leftarrow_R \{1, 2, \ldots, k\}$
  - $R \leftarrow_R \{0,1\}^{n-v-1}$
  - $IV \leftarrow Enc_{K'}(K[l] \,||l||R)$
  - $C = E_K(M, A, IV)$
  - Return C

- $D'(K', \boldsymbol{C}, A)$ Algorithm ($\boldsymbol{C}$ is an indexed array of ciphertexts):
  - For $j = 1, \ldots, |\boldsymbol{C}|$ do
    - $b \,||l||R \leftarrow Dec_{K'}(X(\boldsymbol{C}[j]))$; $K[l] \leftarrow b$
  - For $j = 1, \ldots, |\boldsymbol{C}|$ do
    - $\boldsymbol{M}[j] \leftarrow D_K(C[j], A[j])$
  - Return $\boldsymbol{M}$

Stateless IV Attack



Comments

- Stateless attack requires some amount of messages to recover the key

- $\approx kln(k)$ messages to recover a key with length k
  - $k = 256 \rightarrow \approx 616$ IVs

- Once the key is recovered, can go back and decrypt all previous conversations

Biased Ciphertext Attack

- What if the encryption algorithm does not surface an IV?
  - CBC2, IACBC, XCBC$,…. Do not source IVs

- We can use a biased ciphertext attack instead
  - Let $F_{K'}\{0,1\}^* \rightarrow \{0,1\}$ be a secure PRF
  - Select any randomness in such a way that $F_{K'}(C) = K[j]$ where $K[j]$ is the j'th bit of the key K
  - Let j be some stateful counter σ maintained by Big Brother and Π'
  - This will require k messages for Big Brother to recover K

Biased Ciphertext Attack

- Let k be the size of the key (ex. 128 bits), and let $v = \log_2 k$
- Notation: $K[i]$ refers to a single bit of K at index i

- $E'(\mathrm{K}', \mathrm{K}, \mathrm{M}, \mathrm{A}, \sigma)$ Algorithm:
  - $j \leftarrow \sigma \; mod \; k; j \leftarrow j + 1$
  - While True
    - $\delta \leftarrow_R; C = E_K(M, A, \delta)||\sigma$
    - If $F_{K'}(C) == K[j]$ break
  - $C = E_K(M, A, \delta)$
  - Return C

- $D'(K', \boldsymbol{C}, \boldsymbol{A})$ Algorithm ($\boldsymbol{C}, \boldsymbol{A}$ is an indexed array of ciphertexts and associated data):
  - For $j = 1, \ldots, |\boldsymbol{C}|$ do
    - $K[j] \leftarrow F_{K'}(C[j])$
  - For $j = 1, \ldots, |\boldsymbol{C}|$ do
    - $\boldsymbol{M}[j] \leftarrow D_K(C[j], A[j])$
  - Return $\boldsymbol{M}$

Comments

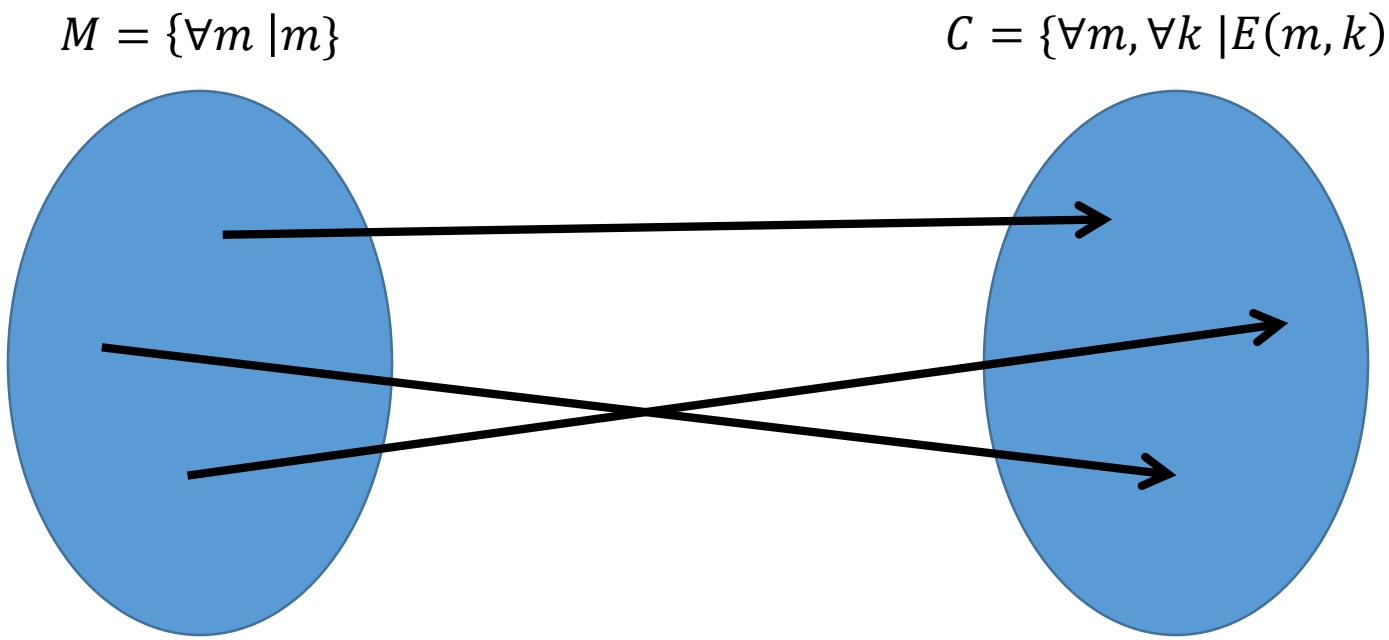
- The biased ciphertext attack will work on every randomized and stateless encryption scheme
  - Why?

Solutions

- These attacks have worked because Π' has a lot of flexibility when it comes to generating its own randomness
- There are several different values of C that will be correctly decrypted by the user, so can select a particular value of C' to leak information
- All of the power of Π' is in its ability to choose a particular C' out of many candidates

An Example Problem – part (a)

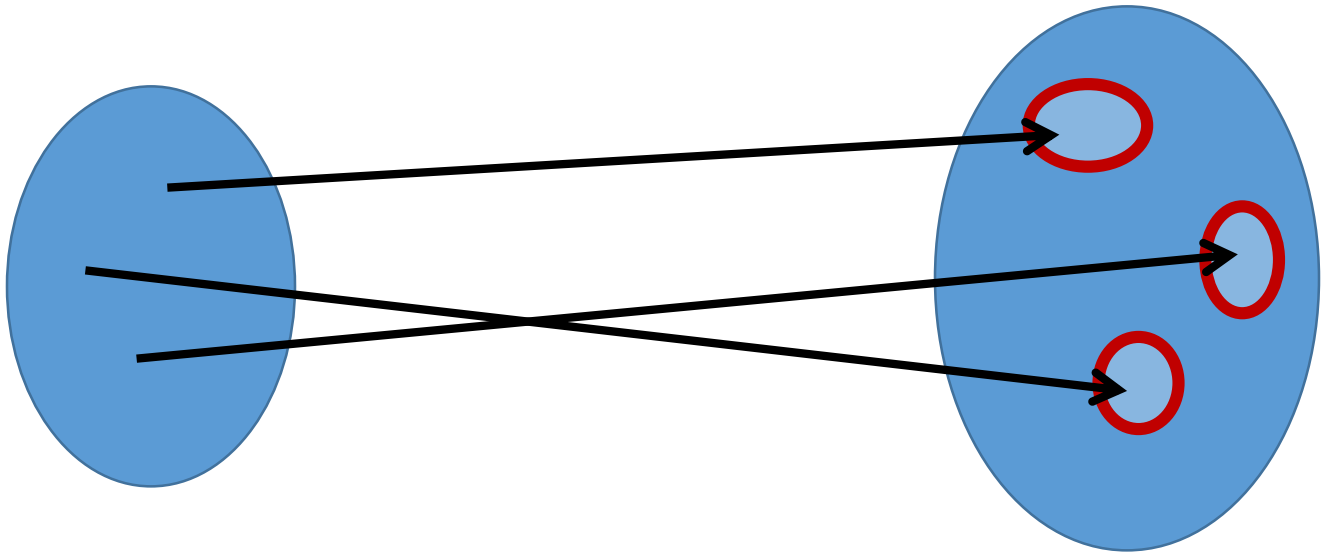
- (E, D) symmetric encryption scheme
- $|E(m,k)| = |m|$: one-to-one mapping between image and preimage



An Example Problem – part (b)

- If $|E(m,k)| = |m| + 1$

$M = \{\forall m\ |m\}$ $C = \{\forall m, \forall k\ |E(m,k)\}$



E is no longer one-to-one. Each pre-image under E has 2^l images

An Example Problem – part (b)

- Question

Suppose for all messages m and keys k, $|E(m,k)| = |m|+l$ for some positive l. Show that an attacker can win the CPA security game if they are allowed to make $2^{l/2}$ queries. This result demonstrates that the keys for such encryption schemes have a finite number of uses before they must be changed.

- Hints
  - Each plaintext m maps into 2^l ciphertexts in average
  - Birthday Bound
  - The advantage of this adversary should be close to 1/2
- Solution?

Unique Ciphertexts

- If Π were such that, given a particular state τ of the user's decryption D, there existed a single unique C such that $D_K(C) = M$, then we would say Π has unique ciphertexts

  - Corollary: If Π' wanted to send the encryption of a particular message M to a user U running D_{K} with internal state τ, then Π' would have no freedom to pick which ciphertext C to send, since there is a unique ciphertext that will work

  - Corollary: E_K is deterministic since it can only produce a single unique ciphertext in this case

  - Corollary: E_K must be stateful and keep state τ

Unique Ciphertext Defense

- $\mathrm{E}(\mathrm{K}, \mathrm{M}, \mathrm{A}, \sigma)$ Algorithm:
  - If $\sigma = 2^l$ return $(\perp, \sigma)$
  - $K_1 || K_2 \leftarrow K$
  - $W \leftarrow P(K_1, <\sigma> || M)$
  - $T \leftarrow F(K_2, W||A)$
  - $C \leftarrow (W, T)$
  - $\sigma \leftarrow \sigma + 1$
  - Return (C, σ)

- $D(K, \boldsymbol{C}, \boldsymbol{A}, \tau)$ Algorithm
  - If $(\tau\ \ 2^l)$ then return $(\perp, \tau)$
  - $K_1 || K_2 \leftarrow K; (W, T) \leftarrow C; x \leftarrow P^{-1}(K_1, W)$
  - If $(|x| < l)$ then return $(\perp, \tau)$
  - $<\sigma> || M \leftarrow x$
  - If $\left(T \neq F(K_2, W||A)\right)$ then return $(\perp, \tau)$
  - If $(\sigma \neq \tau)$ then return $(\perp, \tau)$
  - $\tau \leftarrow \tau + 1$; Return (M, τ)

P: family of keyed permutations, F: family of keyed functions

- It can be shown that Surveillance game advantage of B is zero!

Other Kinds of Attacks

- ASA's on public key encryption
  - RSA, DH, Signature Schemes
  - Elliptic Curves

- Attacks on hardware implementation of crypto
  - Intel AES instruction set
  - Specialized hardware

- Attacks on compilers / runtime state

- Side channel attacks using timing information

- Etc.

Family of Elliptic Curves

- $G = \left\{ \begin{matrix} (x, y) \in (\mathbb{F}_P)^2 \colon y^2 = x^3 + ax + b \ (mod\ p) \\ \wedge\, 4a^3 + 27b^2 \neq 0 \ (mod\ p) \end{matrix} \right\} \cup \{0\}$

- Suppose a user U contacts a server, and the server suggests that they do key exchange ($ECDH$) using a particular curve from the family G where $a = 71817391928581013 8$ and $b = 1213401234871075 6$

- How should the user feel about this? Could it be the case that the server actually found a clever attack and then picked the group parameters specifically so that the attack would work?

- Could it be the case that standardizations of elliptic curve groups are of this form?

Family of Elliptic Curves

- Idea: Instead of just suggesting (a, b), suggest (s, a, b) where $(a, b) = H(s)$ and H is a collision resistant hash function

- Now an (a, b) cannot be chosen directly, instead an s must be chosen such that its hash is (a, b). If there was a very clever attack against the group with a particular (a, b), the collision resistant and 1-way properties of a hash function make it difficult to find s' such that $(a, b) = H(s')$

- Could still be the case that a large family of parameters (a, b) are vulnerable, and select s randomly until it hashes to a vulnerable group

Big Key Cryptography

- PRGs can expand a small seed into a large string of pseudo random bits (say, 2^{80} bit long)
- Idea: Use the entire output of the PRG as a key
- Corollary: Nobody can steal the key because nobody can even read the entire thing!

Big Key Cryptography

- Every time a user U wants to send a message, he computes $R \leftarrow_R \{0,1\}^{128}$ (128 is a parameter)
- Compute $\mathrm{I}[1] = H(R,1),\ I[2] = H(R,2)$ …
- Compute key as $K_s = H\big(R, K\big[I[1]\big], K\big[I[2]\big], \dots\big)$ where K is a key of very long length, like 2^{80} or 2^{128}
- Need a way to compute an index of K without computing the whole thing
  - Blum-Blum-Shub PRG has this property

Thanks!