Information security

Digitalization in public life

- Internet access - WiFi and mobile networks
- Accounts: Ashoka, Google, Instagram, Facebook,…
- Banking, UPI, Credit cards,
- National identity
- Welfare
- National population and voter registry
- National Digital Health Mission (health registry)
- Public credit registry, Income and other tax registries
- Electronic voting
- Biometric (FR) based access control and surveillance
- Electronic contact tracing: Aarogya Setu
- NATGRID and other surveillance
- AI
- …

Security

- What and why?
- Security vs privacy
- Hardware? Software? Network? Use cases?
- How do we know that a protocol is secure? How do we analyse security?
- Does crypto give us security?
    - Software?
    - Key?
    - Protocol?

Nature of informational privacy

Digital Person - Daniel J Solove, Supreme Court

- **Orwellian dangers**: surveillance state; big brother; panopticon
- **Secrecy paradigm**: harm occurs when one's hidden world is uncovered to the public
- **Invasion paradigm**: intrusion into one's private world can cause harm; such as with linking of data points
- **Kafkaesque dangers**: insensitive, opaque, and uncontrollable bureaucracy; helplessness and vulnerability of individuals; dehumanisation; AI (bias and fairness)

Basic digital presence

- Authentication
- Encryption
- Digital signature
- Trust points ?

Crypto basics: symmetric key

- Alice (A) and Bob (B) have a pre-shared key K. Only they have K
- A encrypts a message M to generate cipher text C using K. We denote this as

$$C = \{M\}_K$$

- B decrypts using K^{-1}

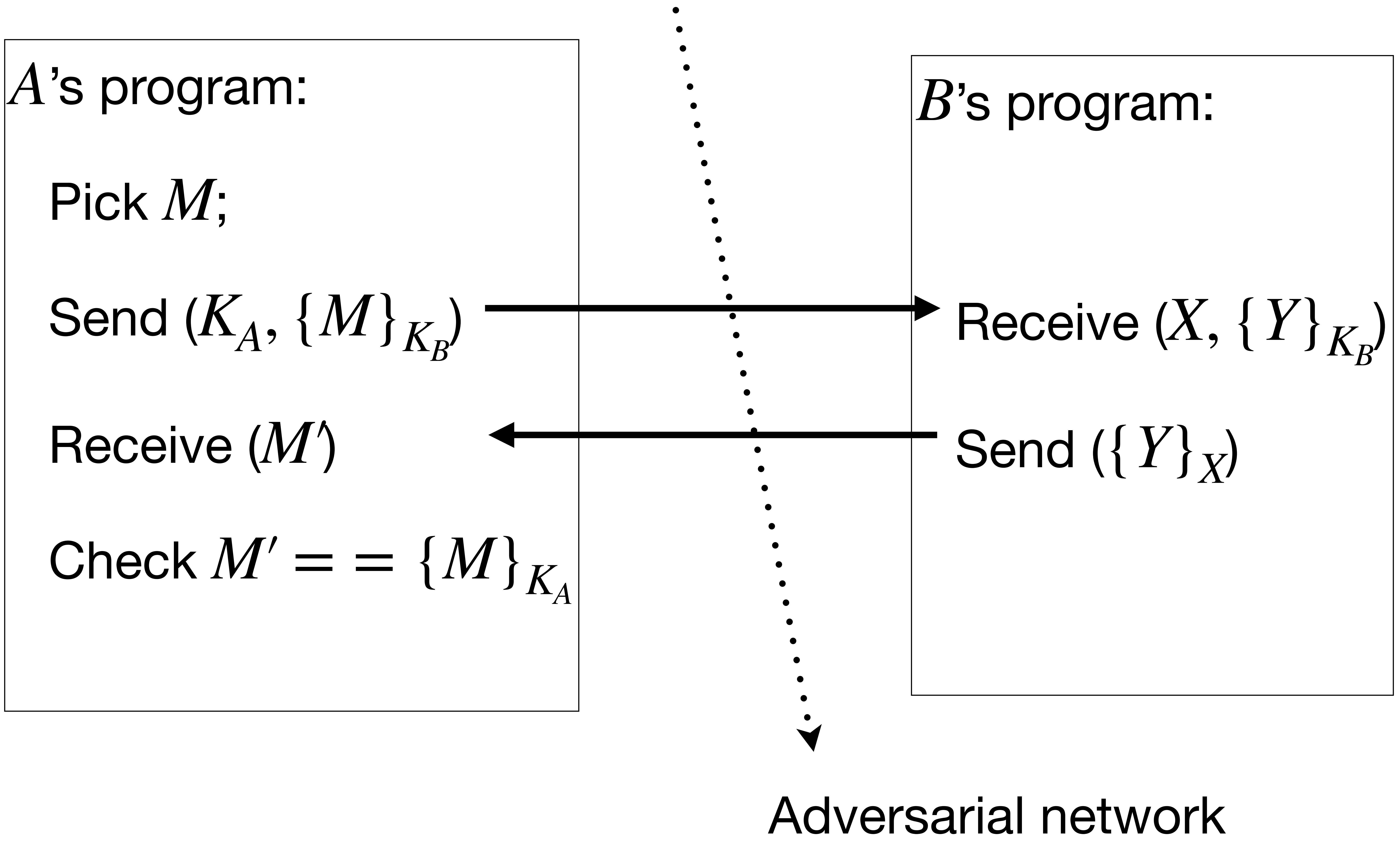
$$M = \{C\}_{K^{-1}}$$

- Example: *Substitution ciphers*. Attacks?

Crypto basic: public key cryptography

- Both A and B have public-secret key pairs (K_A, K_A^{-1}) and (K_B, K_B^{-1})
- K_A and K_B are public information, K_A^{-1} and K_B^{-1} are secret info of A and B
- For both, $C = \{M\}_K \iff M = \{C\}_{K^{-1}}$
- To **encrypt a message** M for B, A sends $C = \{M\}_{K_B}$. Only B can decrypt with $M = \{C\}_{K_B^{-1}}$
- To **sign a message** M, A computes $M' = \{M\}_{K_A^{-1}}$ and sends (M, M'). Anybody can verify $\{M'\}_{K_A} = M$.
- A can combine the above two to send a signed and encrypted message to B (**figure out how and submit by EOD**)

A crypto protocol



Secure?

- **After a valid execution, nobody other than A and B should know M**
- Does the above always hold? Assume the crypto is *bulletproof*
- Suppose Eve (E) is a *(wo)**man in the middle***
- A sends $(K_A, \{M\}_{K_B})$
- E captures and sends $(K_E, \{M\}_{K_B})$ to B
- B sends back $\{M\}_{K_E}$. E captures. Gone!
- E sends back $\{M\}_{K_A}$ to A. A*'s check passes.*

Threat models

Basics of threat modelling

- Threat actors
- Adversaries
- Capabilities of adversaries and system properties
- Trust vs verifiability
- Clear articulation of all trust points

Case study: Authentication and KYC

Trust model of old-fashioned identity cards

- Presenter trusted?
- Verifier trusted?
- KYC based on identity documents?

Trust model of old-fashioned identity cards

- Presenter trusted?
- Verifier trusted?
- KYC based on identity documents?
    - Possibilities of repurposing?
- Vacuous?

Trust model of smart cards with chips

- Content trustworthy?
    - Under what conditions?
- Presenter?
- Verifier?
- Verifier machine?

Trust model of Aadhaar Based Biometric Authentication

- No trust requirement on presenter?
- What about verifier? Machine?

Trust model of Aadhaar Based Biometric Authentication

- No trust requirement on presenter?
- What about verifier? Machine?
- Assume cannot control backend
    - False authorisation and/or accounting?
    - Store and replay?
- What if authentication outcome is routed through the verifier?

Trust models of other authentication methods?

- Passwords
- Ssh authentication (Diffie-Helman key exchange)
- Kerberos authentication