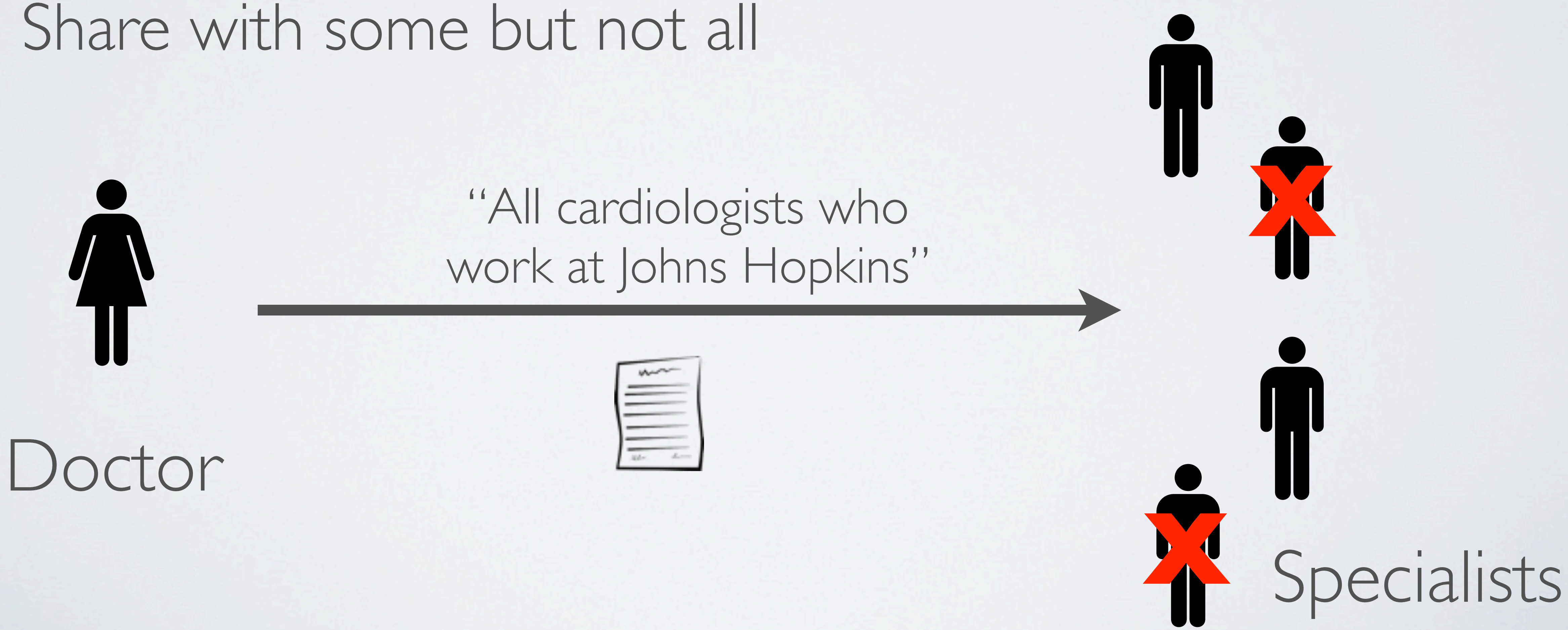
Outsourcing the Decryption of ABE Ciphertexts

Matthew Green and Susan Hohenberger
Johns Hopkins University

Brent Waters
UT Austin

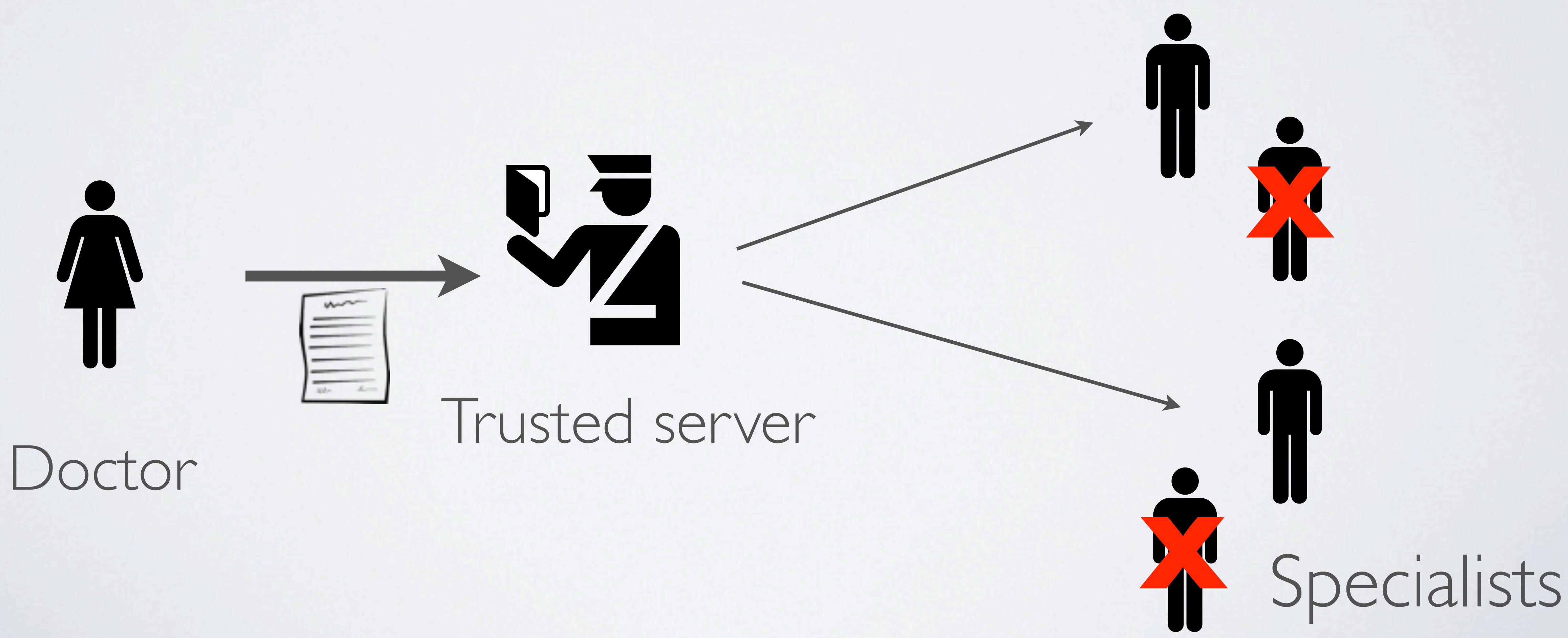
Background

- A problem
  - Securing records in a data-sharing environment
  - E.g., medical records, sensitive documents, etc.
  - Share with some but not all



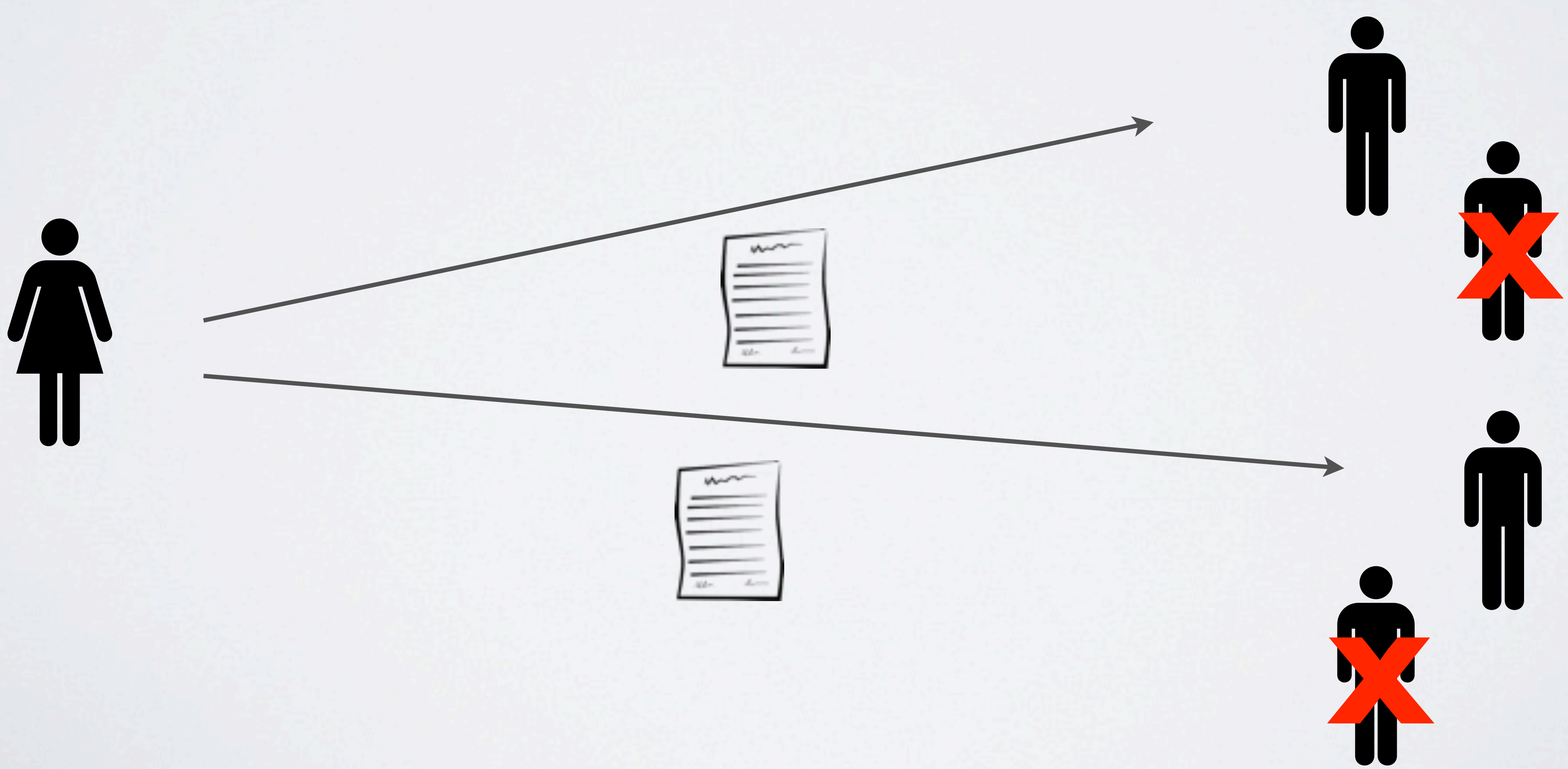
Traditional access control

- Relies on a trusted party (reference monitor)
  - Non-cryptographic
  - Well-known drawbacks: software, insiders, <u>availability</u>



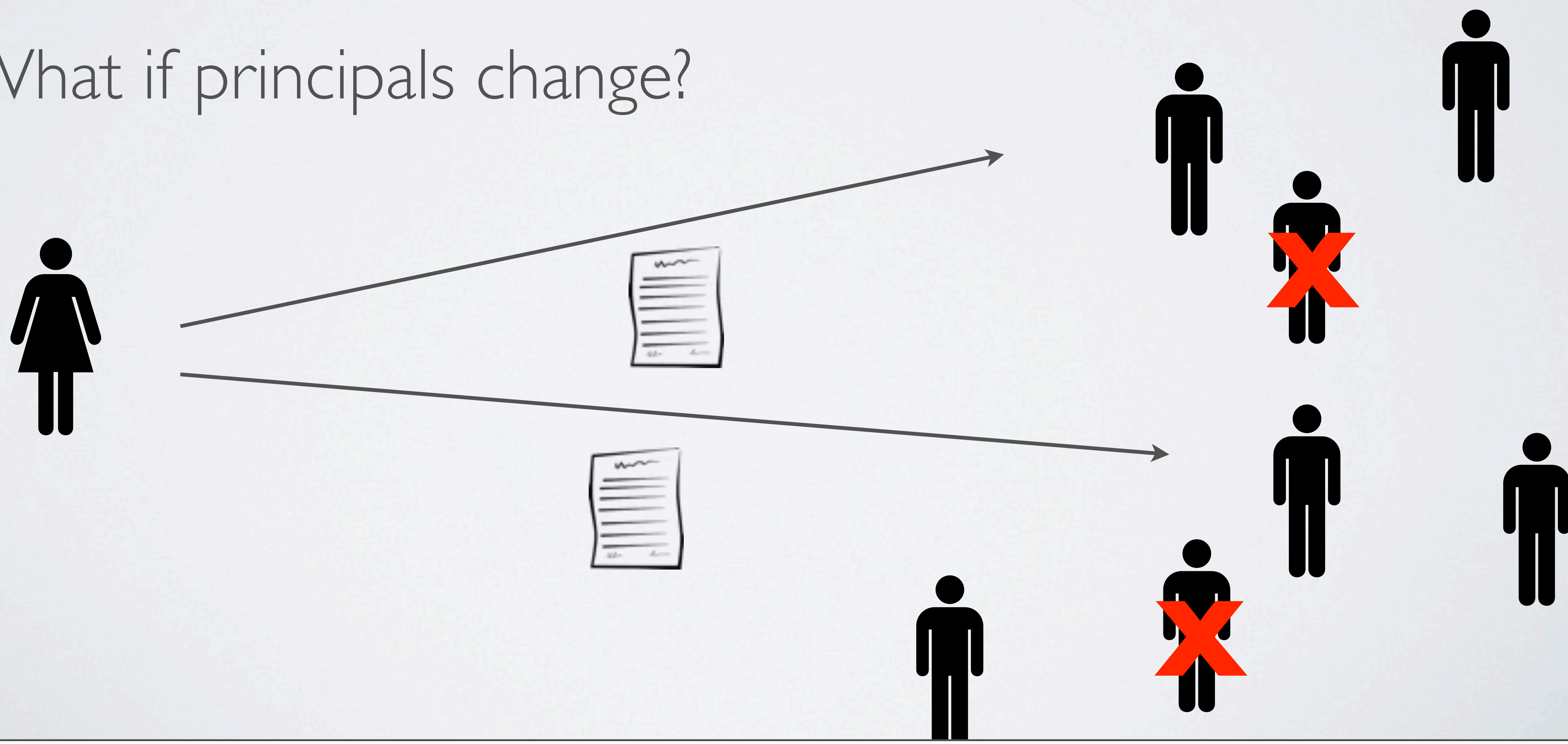
Cryptographic access control

- Traditional approach (public-key encryption)
  - Encrypt record to all valid recipients
  - Problem: must know all possible recipient keys



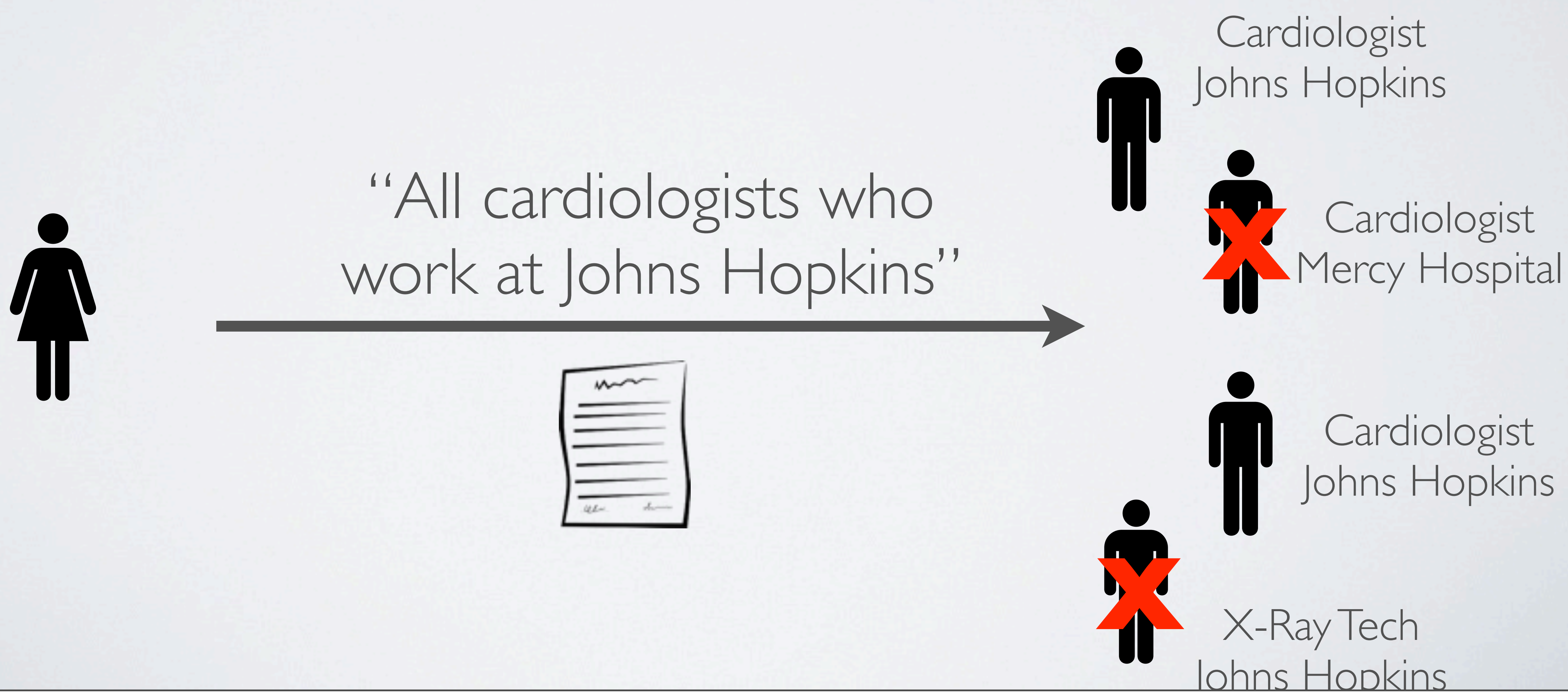
Cryptographic access control

- Traditional approach (public-key encryption)
  - Encrypt record to all valid recipients
  - Problem: must know all possible recipient keys
    - What if principals change?



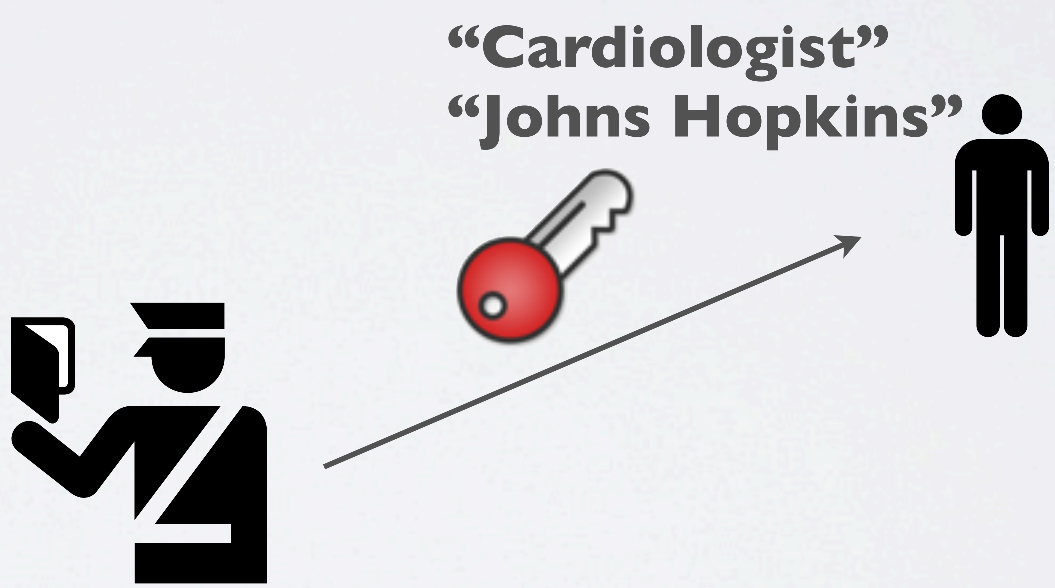
ABE

- Attribute-Based Encryption [Sahai-Waters '05]
  - Extension of Identity-Based Encryption
  - Encrypt to users with certain *attributes*



CP-ABE

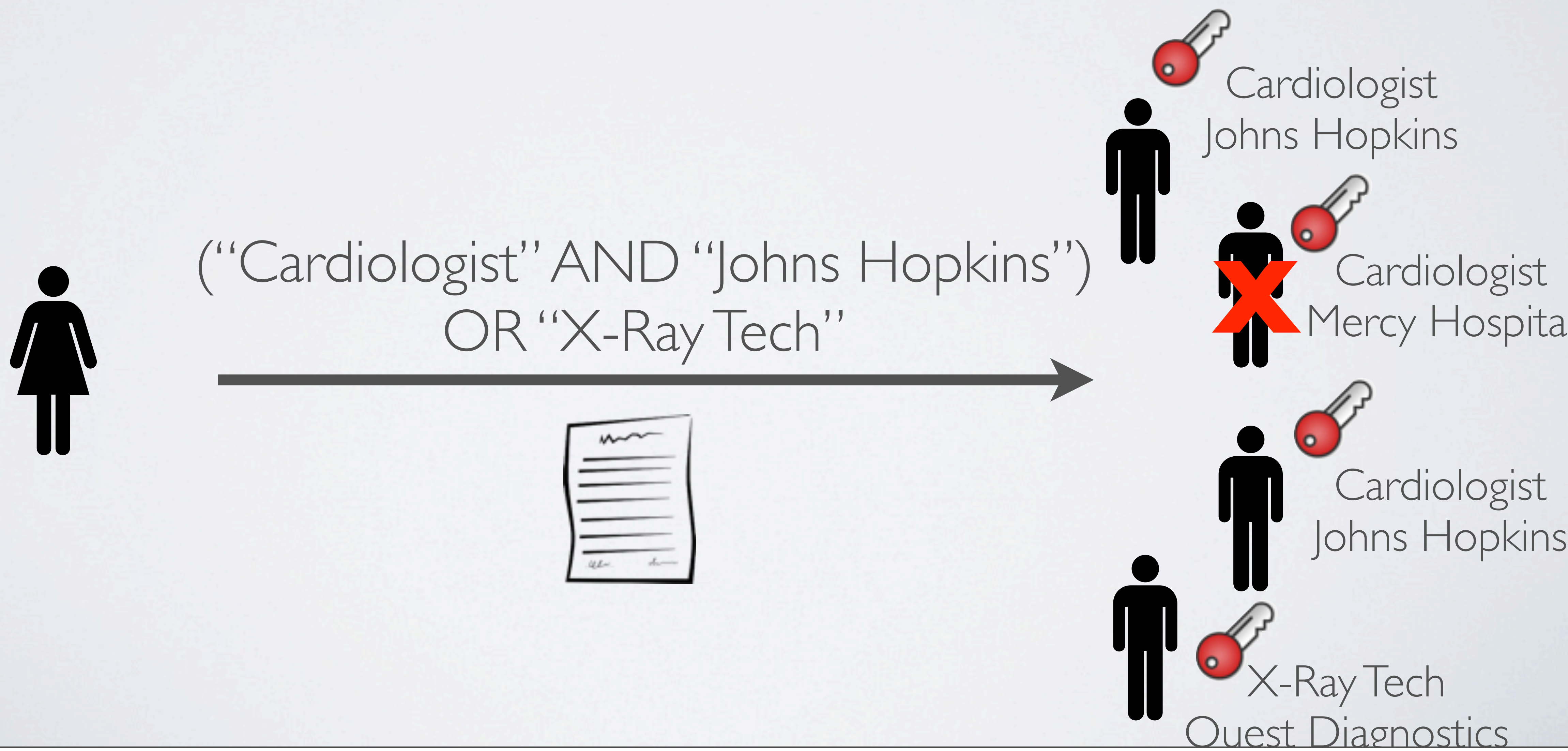
- Ciphertext-policy ABE [BSW07]
  - User secret keys bound to a list of attributes
  - Users obtain keys from an authority



Key Authority

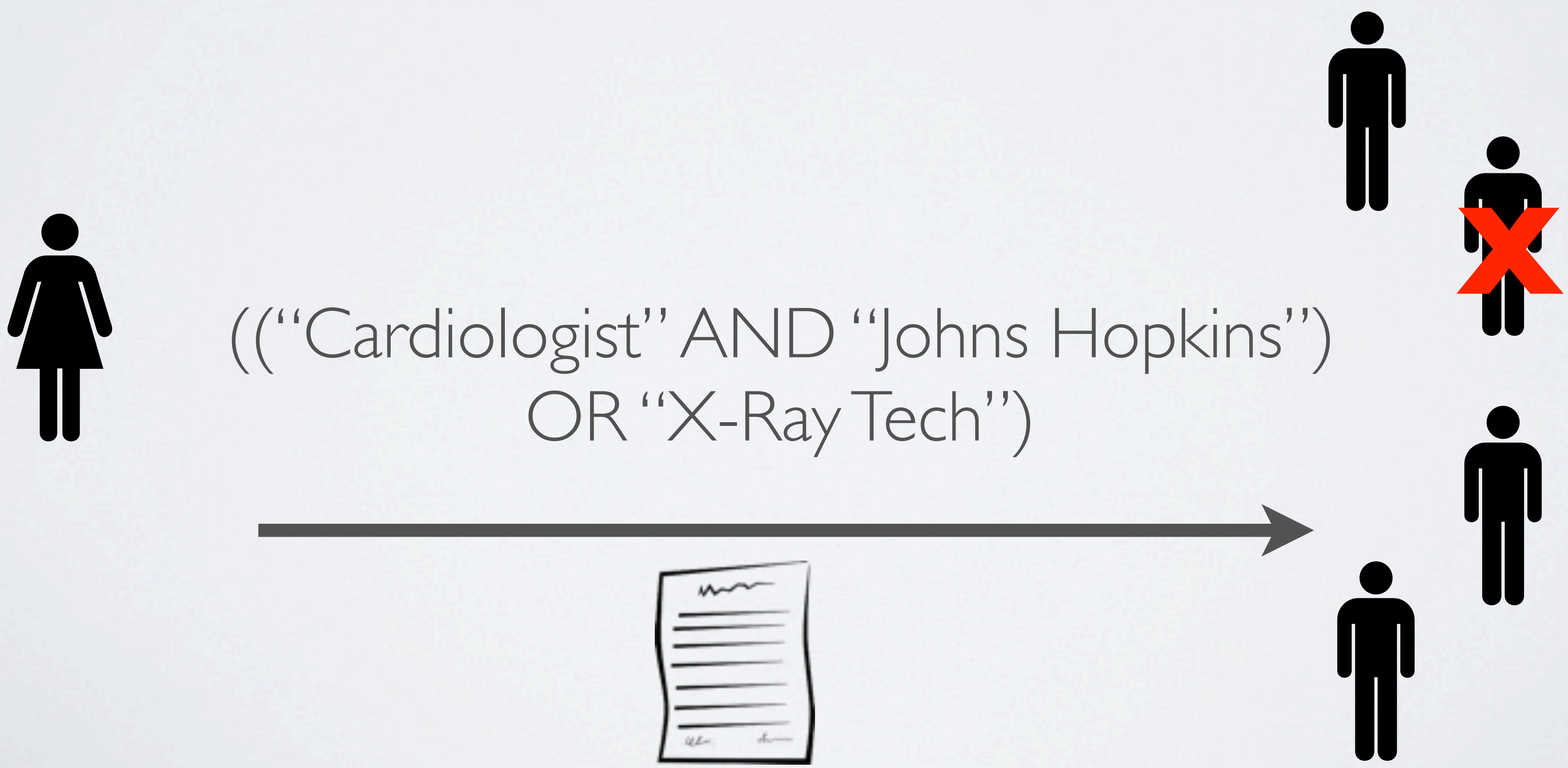
CP-ABE

- Ciphertext-policy ABE [BSW07]
  - Encryptors can specify a policy as a *boolean formula* over attributes



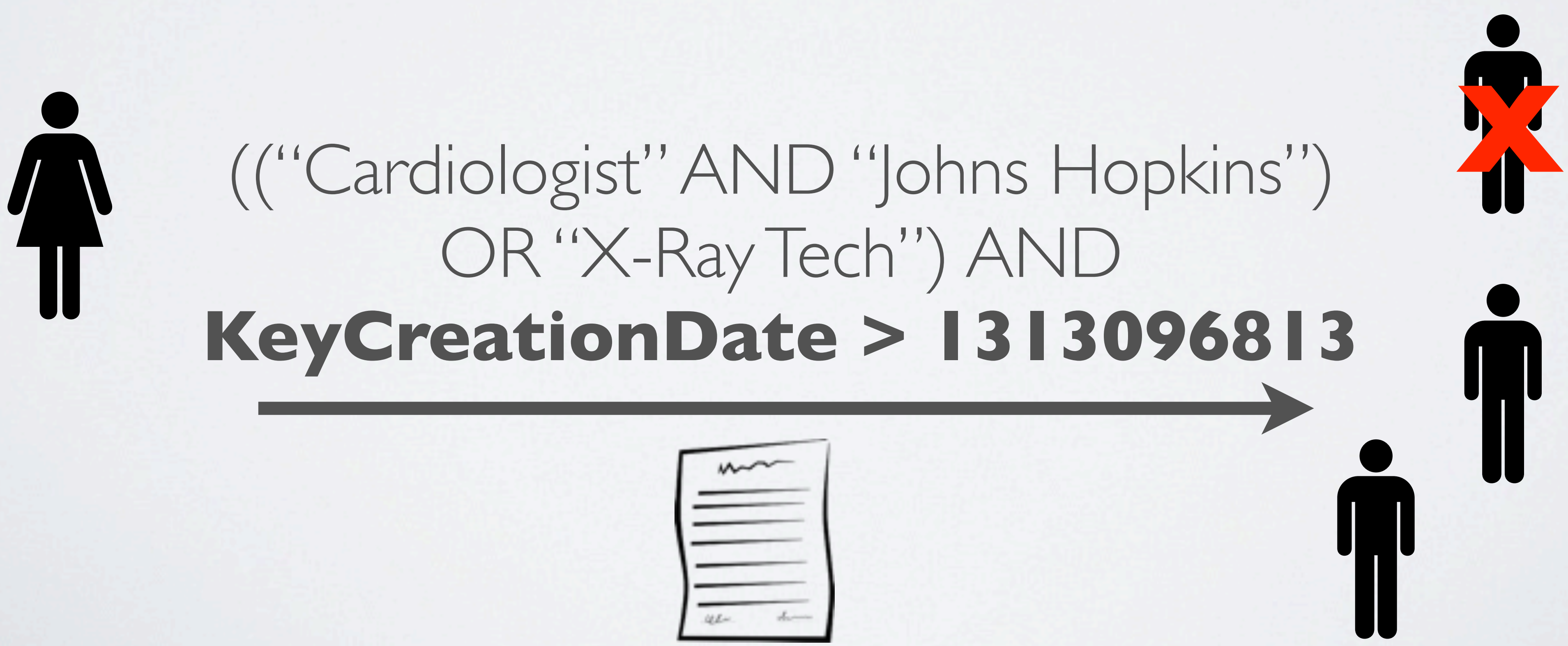
CP-ABE

- Ciphertext-policy ABE [BSW07]
  - Formulae can use arbitrary numbers of AND, OR, (m-of-n Threshold) gates



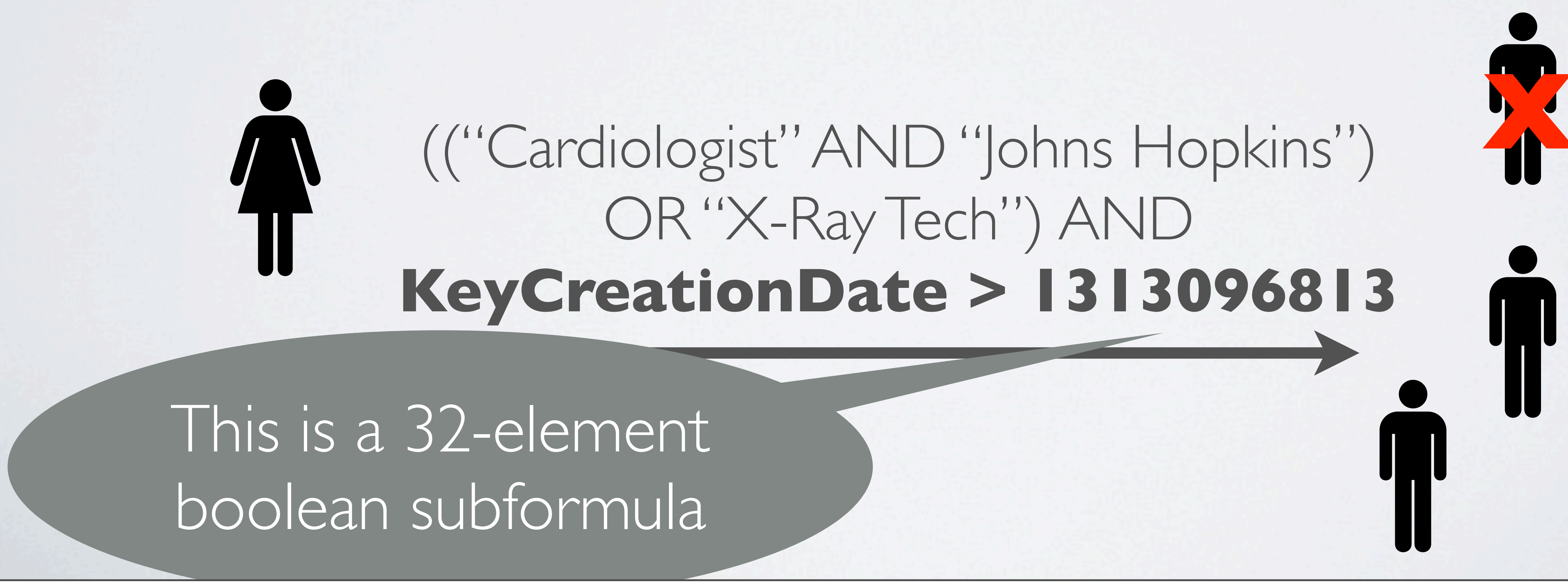
CP-ABE

- Ciphertext-policy ABE [BSW07]
  - Formulae can use arbitrary numbers of AND, OR, (m-of-n Threshold) gates
  - Using these gates we can build <, >, = operators by representing quantities as binary values



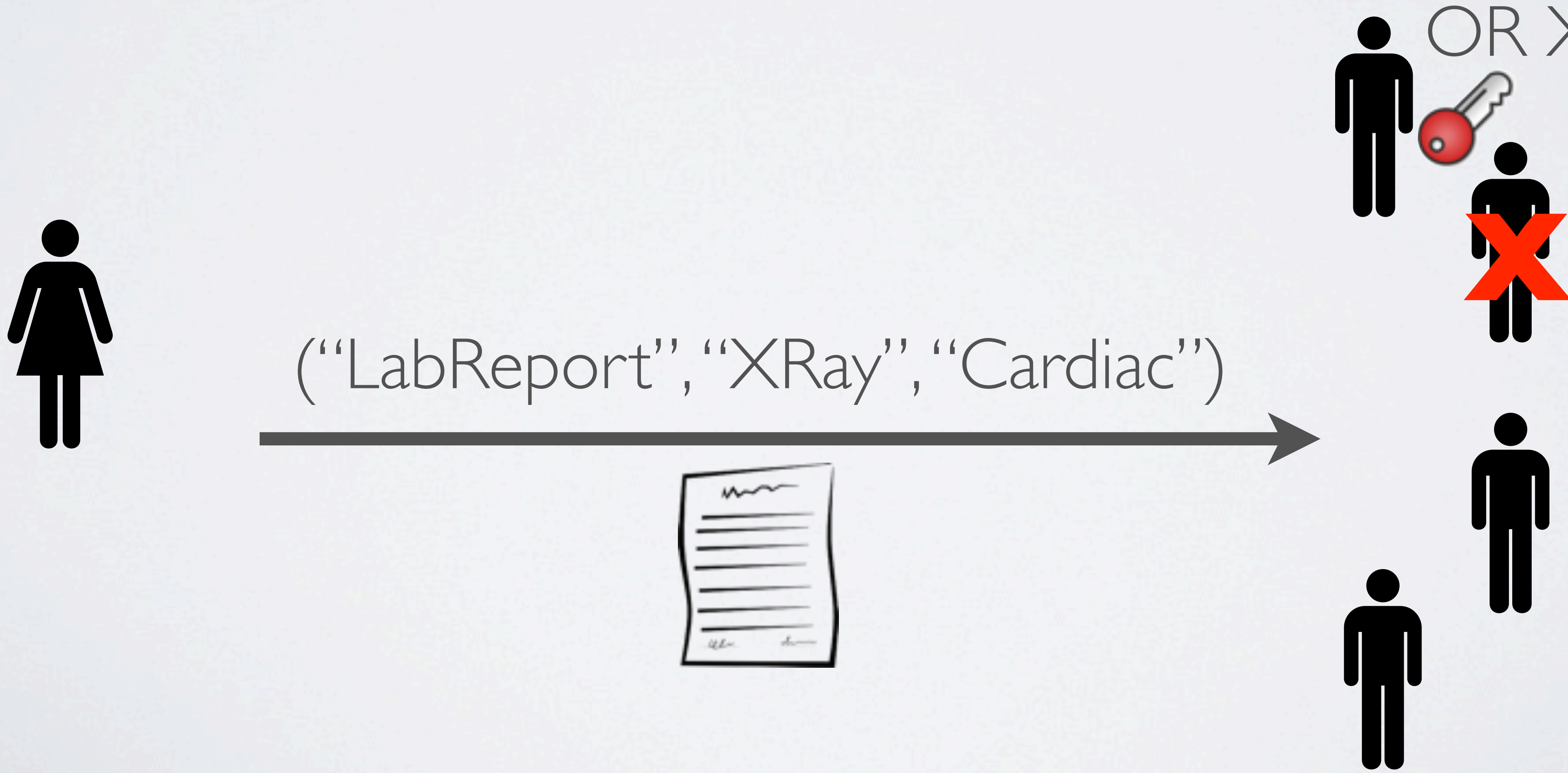
CP-ABE

- Ciphertext-policy ABE [BSW07]
  - Formulae can use arbitrary numbers of AND, OR, (m-of-n Threshold) gates
  - Using these gates we can build <, >, = operators by representing quantities as binary values



KP-ABE

- Key-policy ABE [SW05]
  - All of the same ideas, but policy/attributes are reversed
  - Each ciphertext contains a list of attributes, each key a boolean policy formula



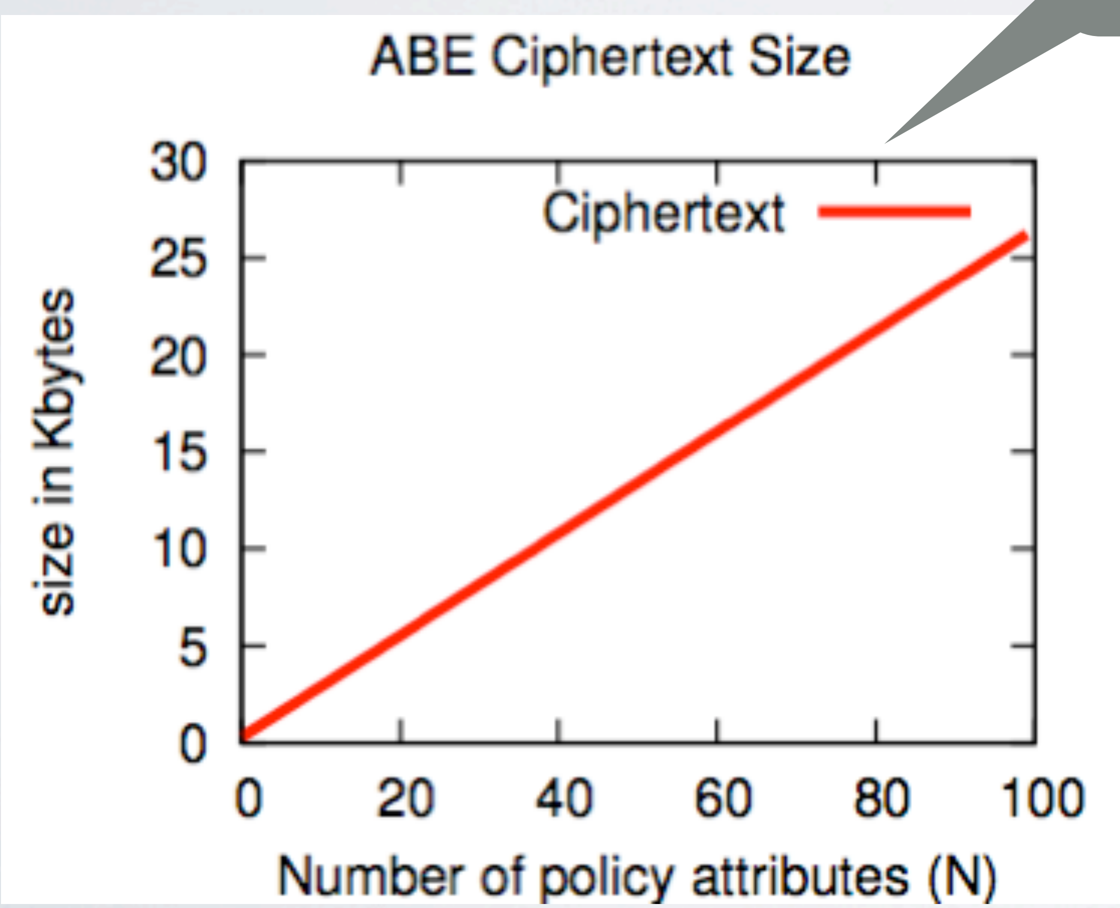
So what's the problem?

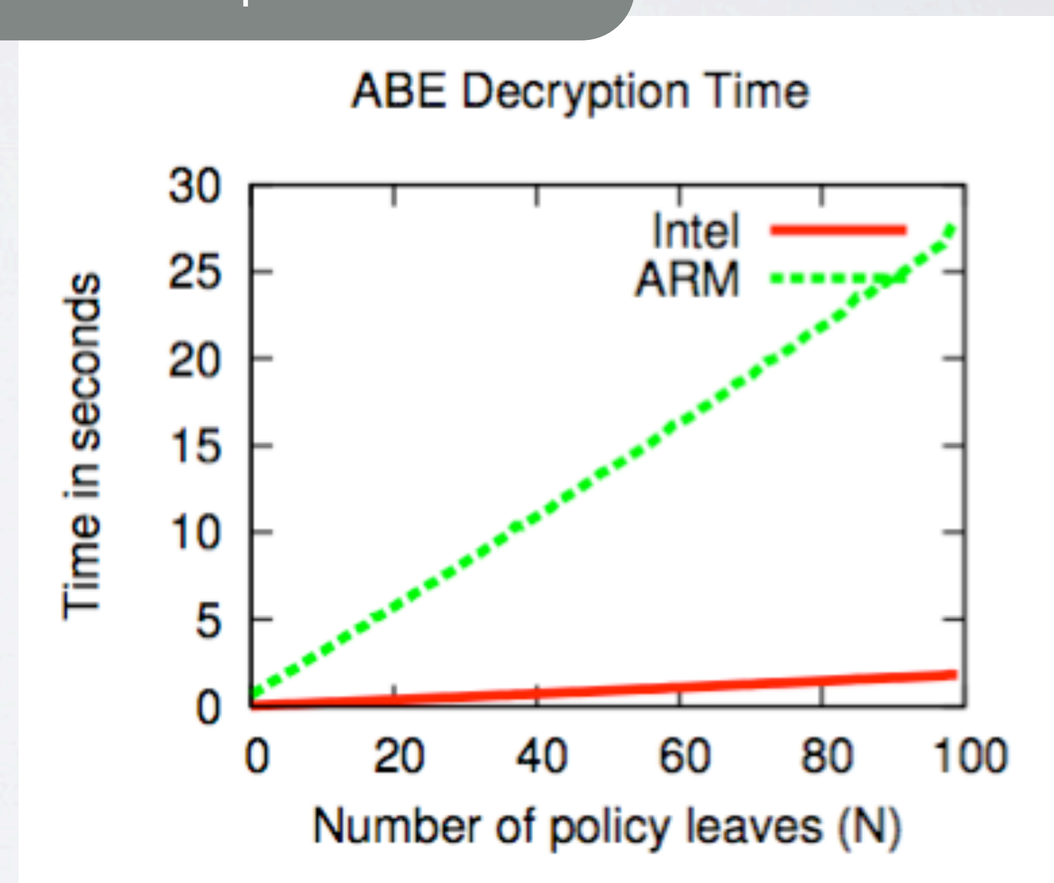
- We have this ABE stuff
  - It lets us implement *arbitrarily* complex encryption policies
  - Doesn't require an on-line reference monitor
  - Why can't we just use it?

So what's the problem?

- Two small wrinkles:
  - Ciphertext size and decryption time grow with the complexity of the access policy (resp. attribute list)

128-bit plaintext





Waters09 CP-ABE scheme, 224-bit MNT curve

So what's the problem?

To the cloud?

Naive Approach

Naive Approach

Remote ciphertext location

Naive Approach

Remote ciphertext location

Plaintexts (smaller)

Naive Approach

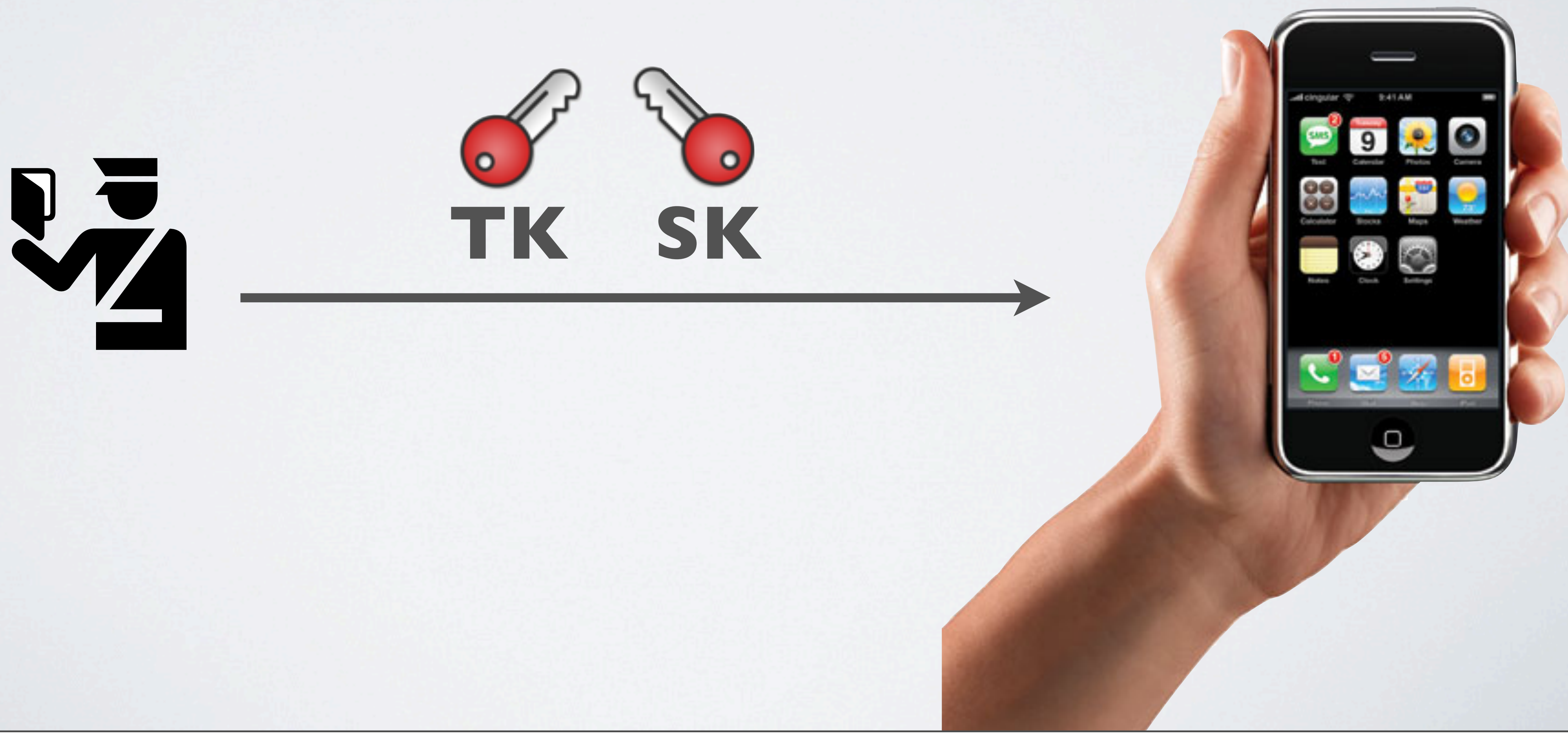
- Problem:
  - We really need to trust the cloud
  - And every fellow cloud user
    - Timing attacks
    - VM exploits
    - CCA attacks

Other approaches

- Why not generic outsourcing techniques?
  - E.g., Craig Gentry's fully-homomorphic encryption
    - This protects the secret key
    - Far too inefficient [GH11]
- Outsourcing pairings [CmCMNS10]
  - Still costly, high bandwidth

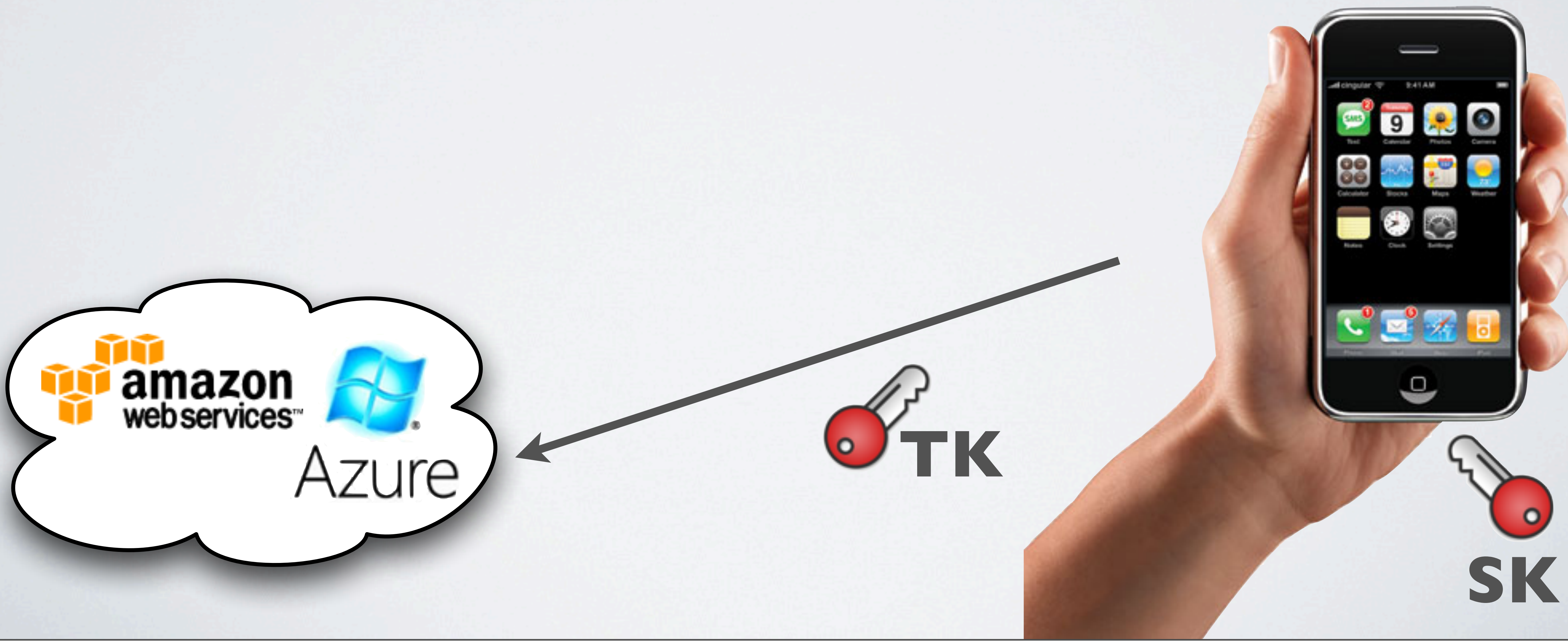
Our Approach

- Change the way that ABE secret keys are generated
  - Authority produces a <u>Transform Key</u> and an Elgamal-style <u>Secret Key</u>



Our Approach

- **TK** can go to anyone (e.g., the cloud)
- Client retains **SK**



Our Approach

- Change the way that ABE secret keys are generated
  - <u>Also define two new algorithms:</u>
    - **Transform**
    - **Dec$_{\mathbf{out}}$**

“Partially-decrypted” ciphertext (smaller!)

$\longrightarrow$ C’

Transform(**TK**, C) -> C’

Dec$_{\mathrm{Out}}$(**SK**, C’)

Outsourcing Security Model

- Traditional CP- (resp. KP-) ABE security def'n:
  - Adversary can query for any secret keys it wants
  - Eventually it asks for a challenge ciphertext on any policy (resp. attr list) not covered by those keys
- New wrinkle:
  - Adversary can query for **TK** on any policy (resp. attr list) <u>with no restrictions at all</u> (i.e., regardless of the challenge)
- This models a *fully adversarial* outsourcing party

Construction: CP-ABE

- Original Waters '09 construction (prime-order bilinear):

$$\text{MPK} = \quad g,\ e(g,g)^{\alpha},\ g^{a}.$$

$$\text{ABE-SK} = K' = g^{\alpha} g^{at} \quad L' = g^{t} \quad \forall x \in S \quad K'_x = H(x)^{t}.$$

Construction: CP-ABE

- Original Waters '09 construction:

$$\text{MPK} = \quad g,\ e(g,g)^{\alpha},\ g^{a}.$$

$$\text{ABE-SK} = K' = g^{\alpha} g^{at} \quad L' = g^{t} \quad \forall x \in S \quad K'_x = H(x)^{t}.$$

$$\Big\downarrow$$

Pick random SK = *z in Zq*

$$\text{TK} = K = K'^{1/z} \qquad L = L'^{1/z} \quad \{K_x\}_{x\in S} = \{K'^{1/z}_x\}_{x\in S}$$

Construction: CP-ABE

- Original Waters '09 construction:

Encryption:

$$C = \mathcal{M} \cdot e(g,g)^{\alpha s},\ C' = g^s,$$

$$(C_1 = g^{a\lambda_1} \cdot F(\rho(1))^{-r_1},\ D_1 = g^{r_1}), \ldots, (C_\ell = g^{a\lambda_\ell} \cdot F(\rho(\ell))^{-r_n},\ D_\ell = g^{r_\ell})$$

Transform:

$$e(C', K) / \left(e(\textstyle\prod_{i \in I} C_i^{\omega_i}, L) \cdot \prod_{i \in I} e(D_i^{\omega_i}, K_{\rho(i)}) \right) =$$

$$e(g,g)^{s\alpha/z} e(g,g)^{ast} / \left(\textstyle\prod_{i \in I} e(g,g)^{ta\lambda_i \omega_i} \right) = e(g,g)^{s\alpha/z}$$

Construction: CP-ABE

- Original Waters '09 construction:

Encryption:

$$\boxed{C = \mathcal{M} \cdot e(g,g)^{\alpha s}},\ C' = g^{s},$$

$$(C_1 = g^{a\lambda_1} \cdot F(\rho(1))^{-r_1},\ D_1 = g^{r_1}), \ldots, (C_\ell = g^{a\lambda_\ell} \cdot F(\rho(\ell))^{-r_n},\ D_\ell = g^{r_\ell})$$

Transform:

$$e(C', K) / \left(e\left(\prod_{i \in I} C_i^{\omega_i}, L\right) \cdot \prod_{i \in I} e\left(D_i^{\omega_i}, K_{\rho(i)}\right) \right) =$$

$$e(g,g)^{s\alpha/z} e(g,g)^{ast} / \left(\prod_{i \in I} e(g,g)^{ta\lambda_i \omega_i} \right) = \boxed{e(g,g)^{s\alpha/z}}$$

,

Construction: CP-ABE

- Original Waters '09 construction:

Encryption:

$$\boxed{C = \mathcal{M} \cdot e(g,g)^{\alpha s}},\ C' = g^s,$$

$$(C_1 = g^{a\lambda_1} \cdot F(\rho(1))^{-r_1},\ D_1 = g^{r_1}), \ldots, (C_\ell = g^{a\lambda_\ell} \cdot F(\rho(\ell))^{-r_n},\ D_\ell = g^{r_\ell})$$

Transform:

$$e(C', K) / \left(e(\textstyle\prod_{i \in I} C_i^{\omega_i}, L) \cdot \prod_{i \in I} e(D_i^{\omega_i}, K_{\rho(i)})\right) =$$

$$e(g,g)^{s\alpha/z} e(g,g)^{ast} / \left(\textstyle\prod_{i \in I} e(g,g)^{ta\lambda_i \omega_i}\right) = \boxed{e(g,g)^{s\alpha/z}}$$

Transformed ciphertext: $e(g,g)^{s\alpha/z}$, $\mathcal{M} \cdot e(g,g)^{\alpha s}$

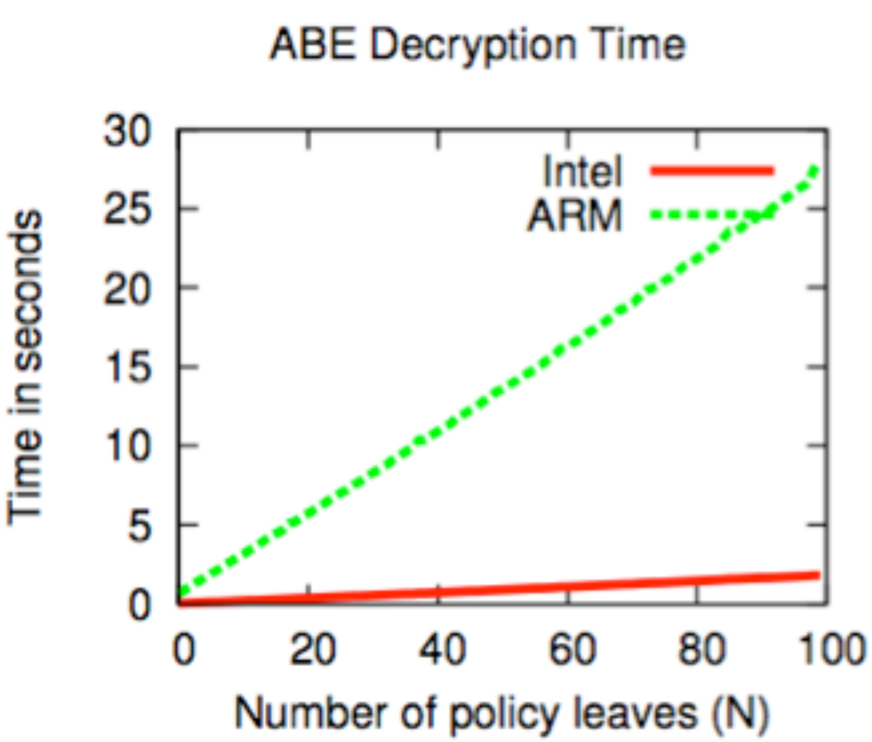
Additional Constructions

- **In the paper:**
  - Security proofs
  - An additional scheme from the Goyal et al. Key-policy ABE [GPSW06]
  - Also: CCA Security for both CP- and KP-ABE (random oracles)

Performance: Waters09

- 3GHz Intel Core Duo, 4GB RAM (one core)
- 412Mhz ARM (iPhone 3G)

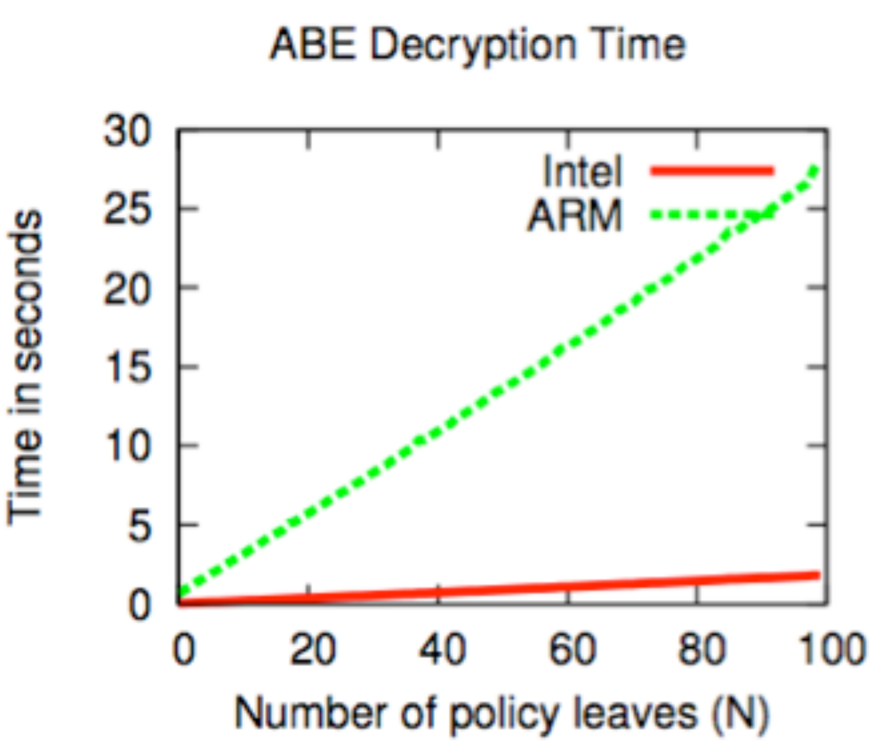
No Outsourcing



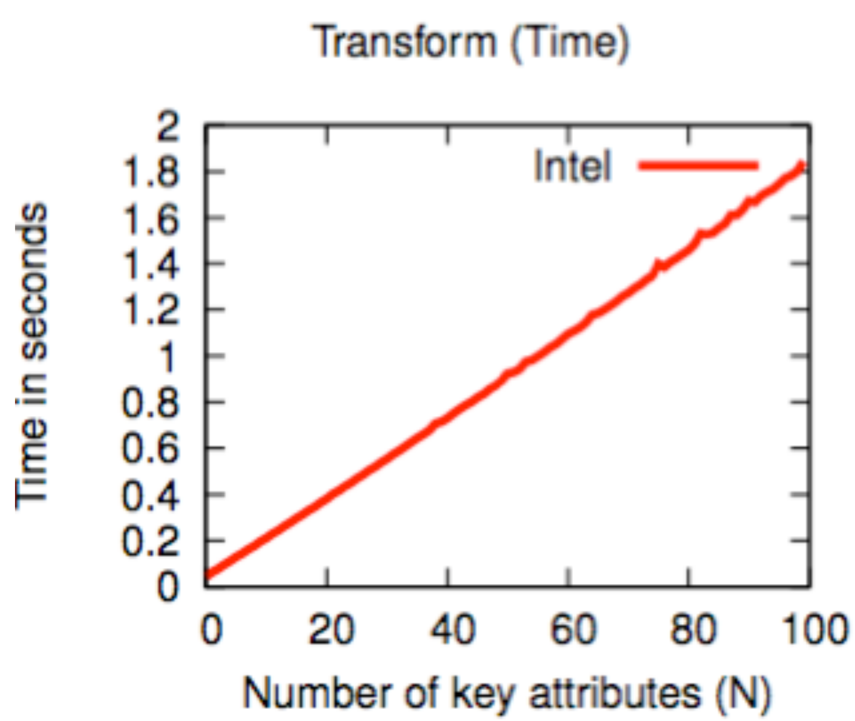
Performance: Waters09

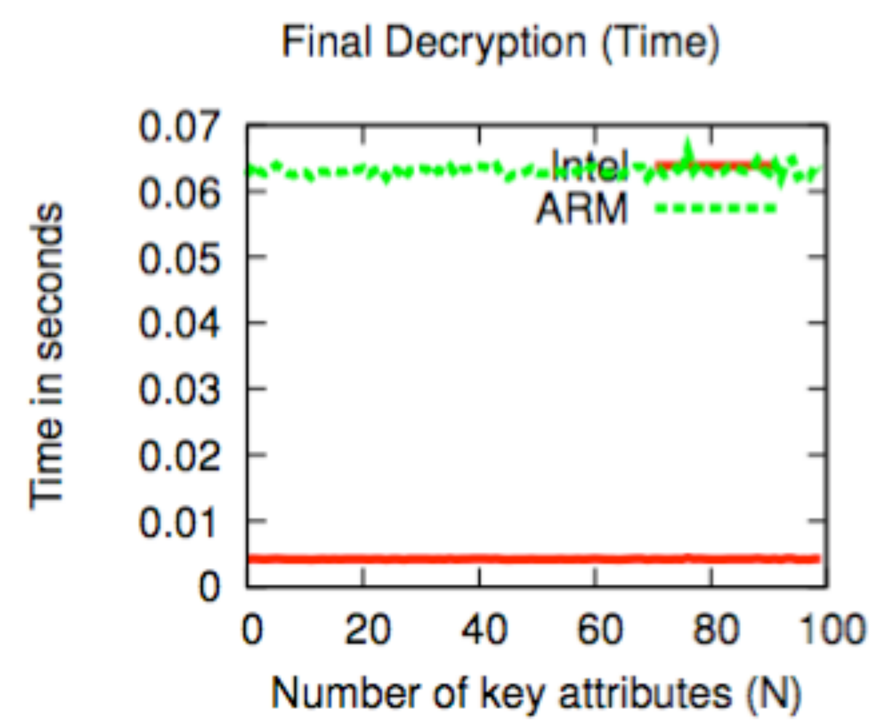
- 3GHz Intel Core Duo, 4GB RAM (one core)
- 412Mhz ARM (iPhone 3G)

No Outsourcing



With Outsourcing

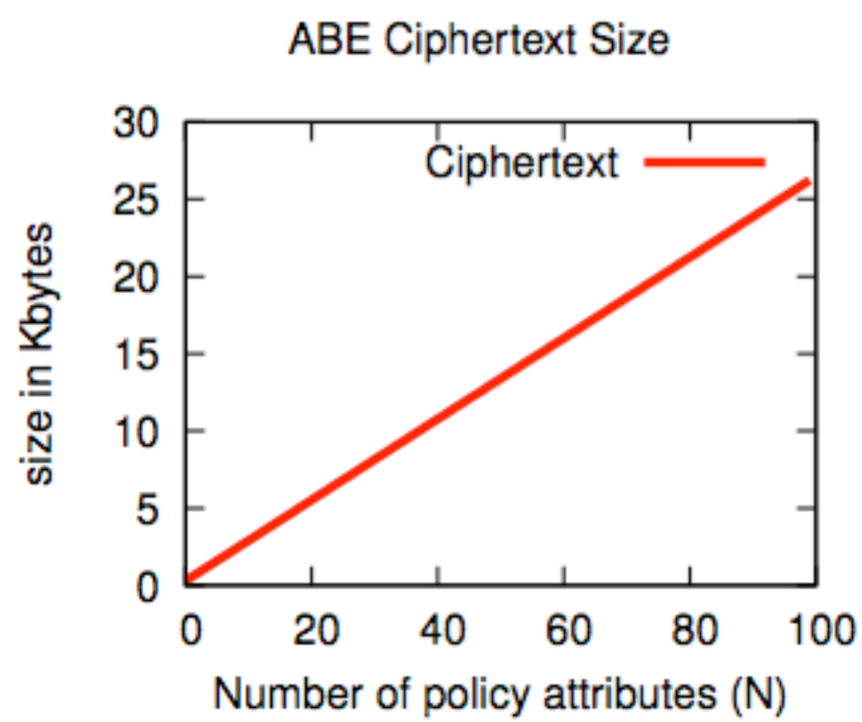


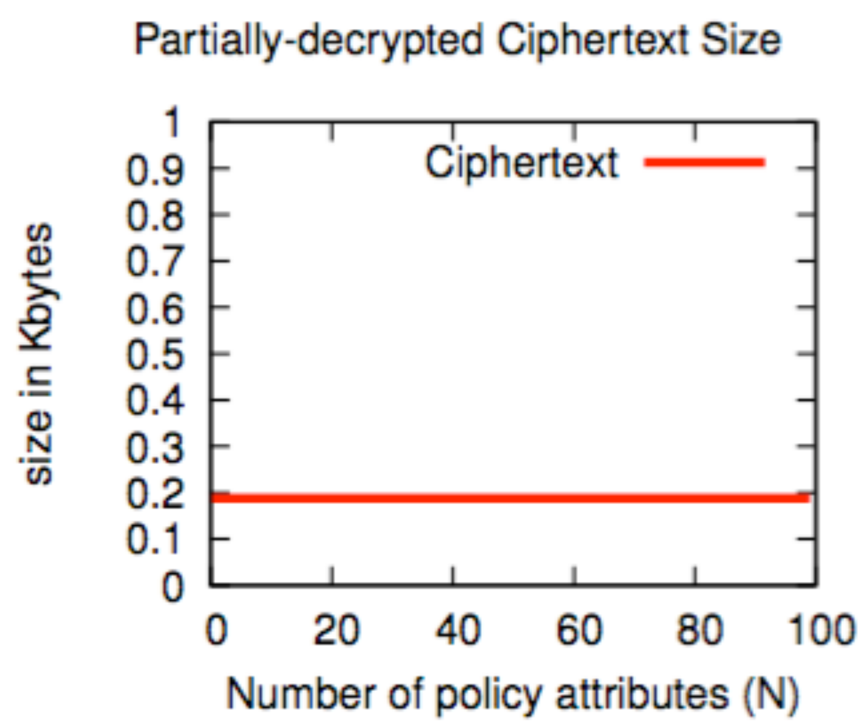


+

Ciphertext Size: Waters09

Before Transform | After Transform





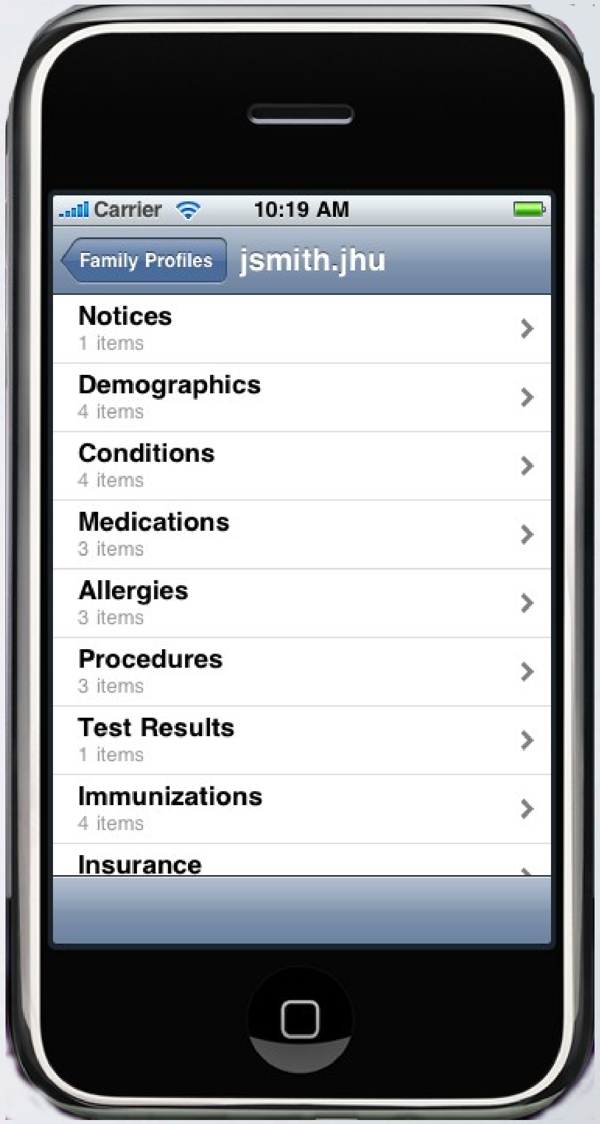
ABE | Elgamal

An EC2-based System

- We constructed Amazon Machine Image ("Proxy") with:
  - Apache
  - Scripts to accept a Transform Key,
    load ciphertexts from remote URLs
  - The code for the Transform algorithm
- Users can programmatically spin up one or more instances

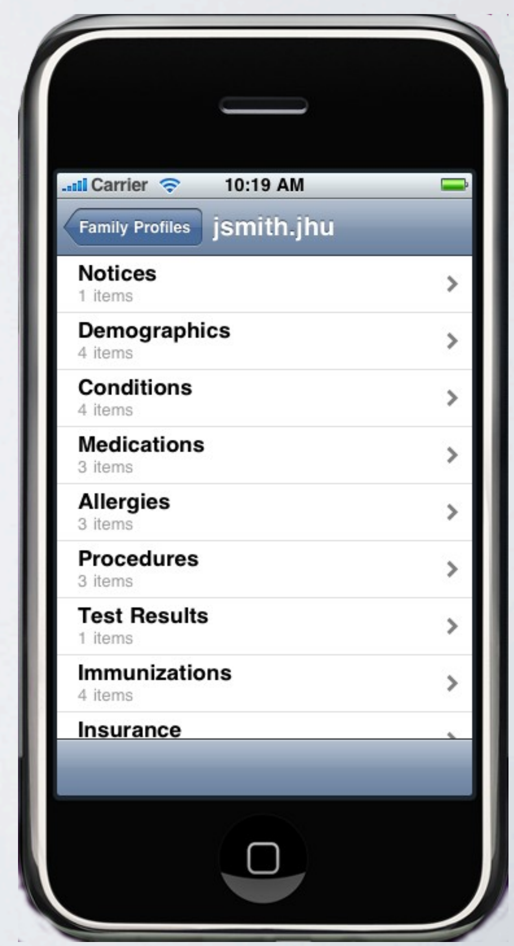
An EC2-based System

- Also created a test application
  - Extended the *iHealthEMR* app from JHU (Ayo Akinyele) (Medical records reader, uses CP-ABE)
  - Added code to transparently instantiate Proxy, upload Transform Key at startup
  - 1-1.5 min for spinup, during which decryption is local.
  - Afterwards it's outsourced!



An EC2-based System

Operation	local-only *(sec)*	local+web *(sec/kb)*	proxy *(sec/kb)*	proxy+web *(sec/kb)*
New proxy instantiation	·	·	93.4 sec	93.4 sec
Restart existing proxy instance	·	·	45 sec	45 sec
Generate & set 70-element transform key	·	·	2.9 sec	2.9 sec
Decryption:				
((DOCTOR OR NURSE) AND INSTITUTION)	1.1s	1.2s/1.1k	.2s/1.4k	.2s/0.4k
(DOCTOR AND TIME > 1262325600 AND TIME < 1267423200)	17.3s	17.3s/22.8k	1.2s/23.2k	1.2s/0.4k



Other Applications

- Outsourcing from smartcards
  - Let the computer do the heavy lifting!
  - Simplify the code base on the smart card

- Reducing TCB
  - ABE implementations are complex:
    parsing code, excess cryptography == vulnerabilities?
  - Let's not trust that all that code:
    - Isolate one trusted piece using e.g., TrustVisor [MLQZDGP10].

Open Problems

- Outsourcing for other cryptosystems (IBE, ABE, NIZKs, Signatures)
- CCA security in the standard model
- A generic cloud-based outsourcing platform
  - Supports many cryptosystems
  - Attacker uploads code of his/her choice at initialization time