RSA Conference2016

San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: CRYP-W03 - Secure Multiparty Computation

Hybrid Publicly Verifiable Computation

James Alderman, Christian Janson, Carlos Cid, Jason Crampton
Royal Holloway, University of London

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Clarus

James Alderman
Post-doctoral Research Assistant
Royal Holloway, University of London
james.alderman@rhul.ac.uk

Contents

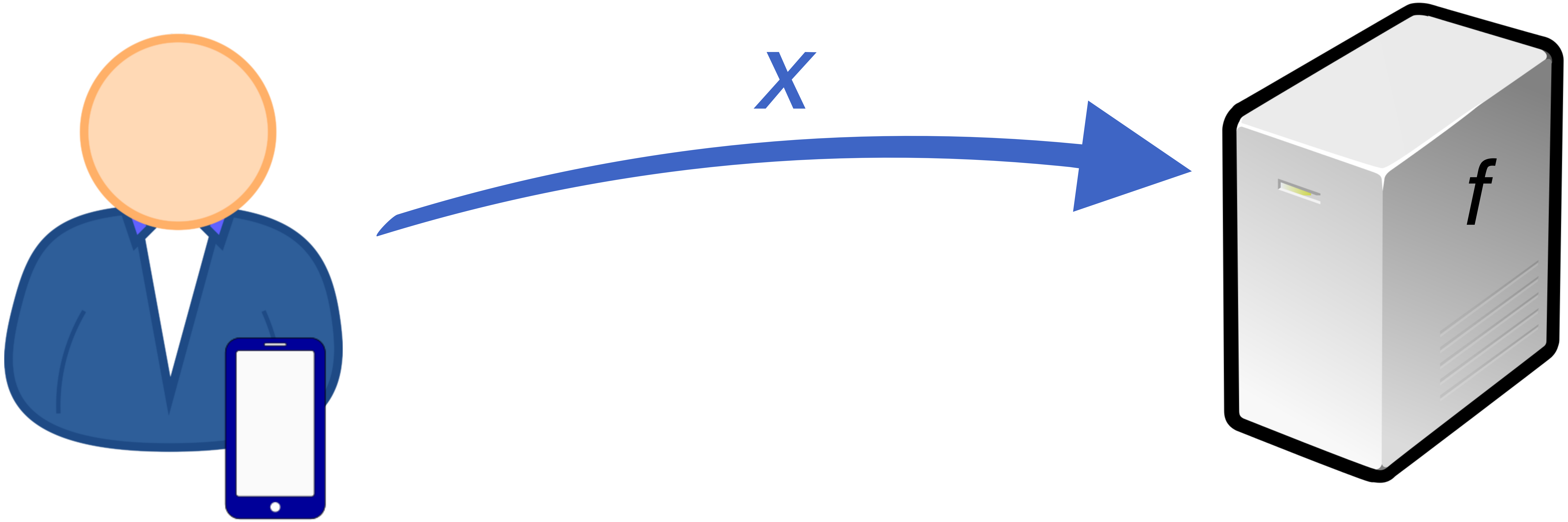
- Background
- Overview of our model
- Technical Details

Background

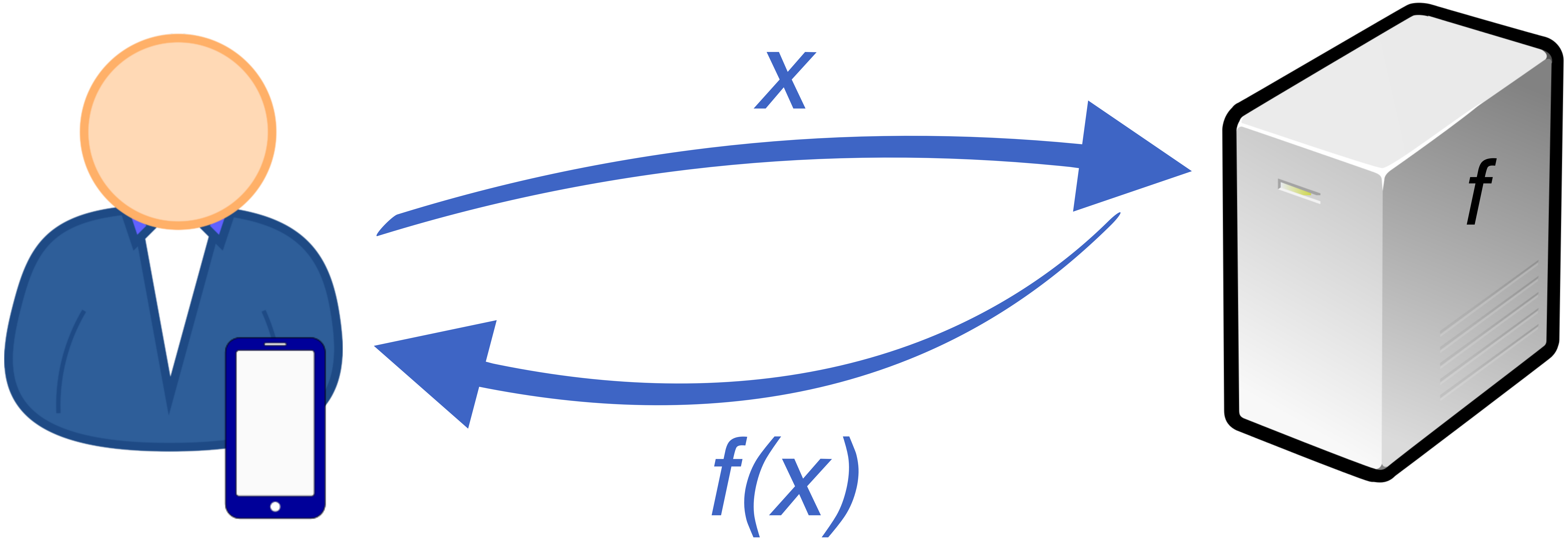
Verifiable Outsourced Computation

Verifiable Outsourced Computation

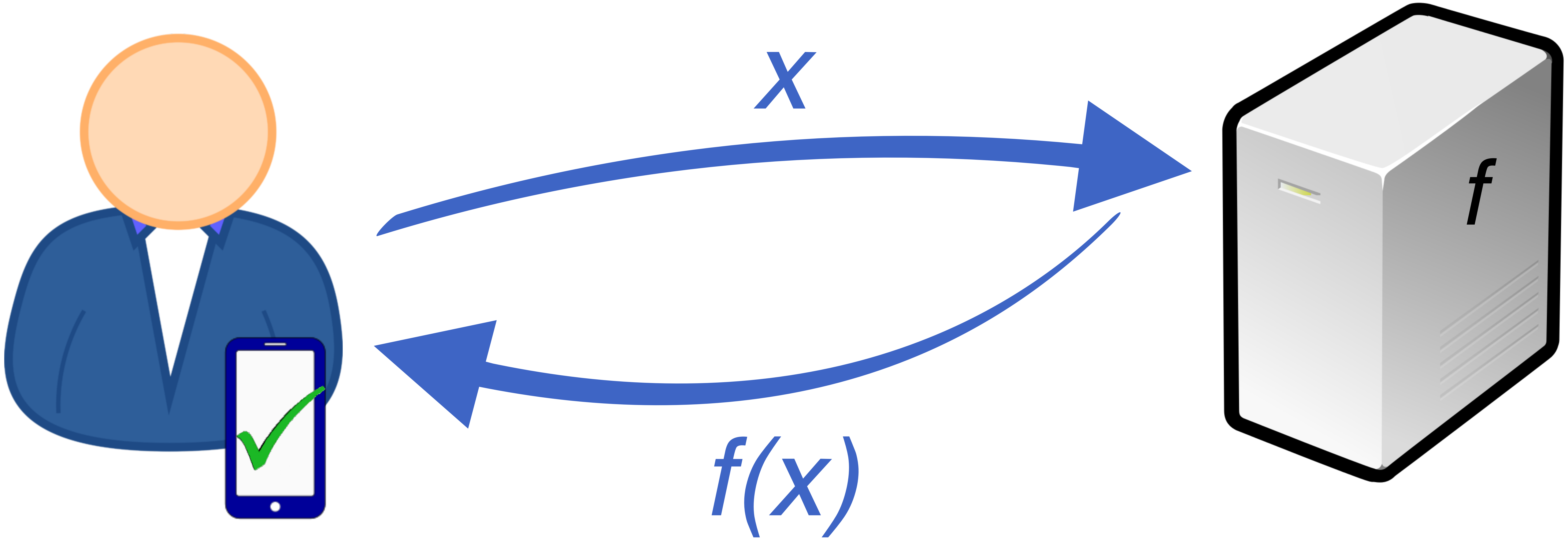
Verifiable Outsourced Computation



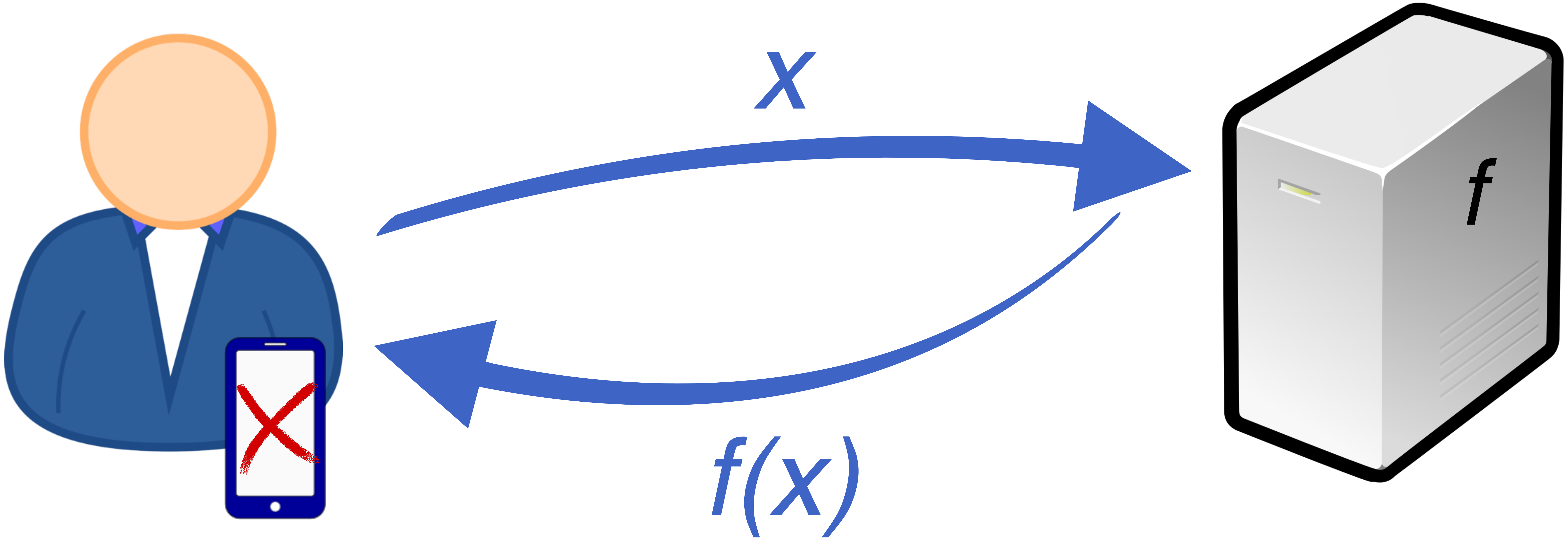
Verifiable Outsourced Computation



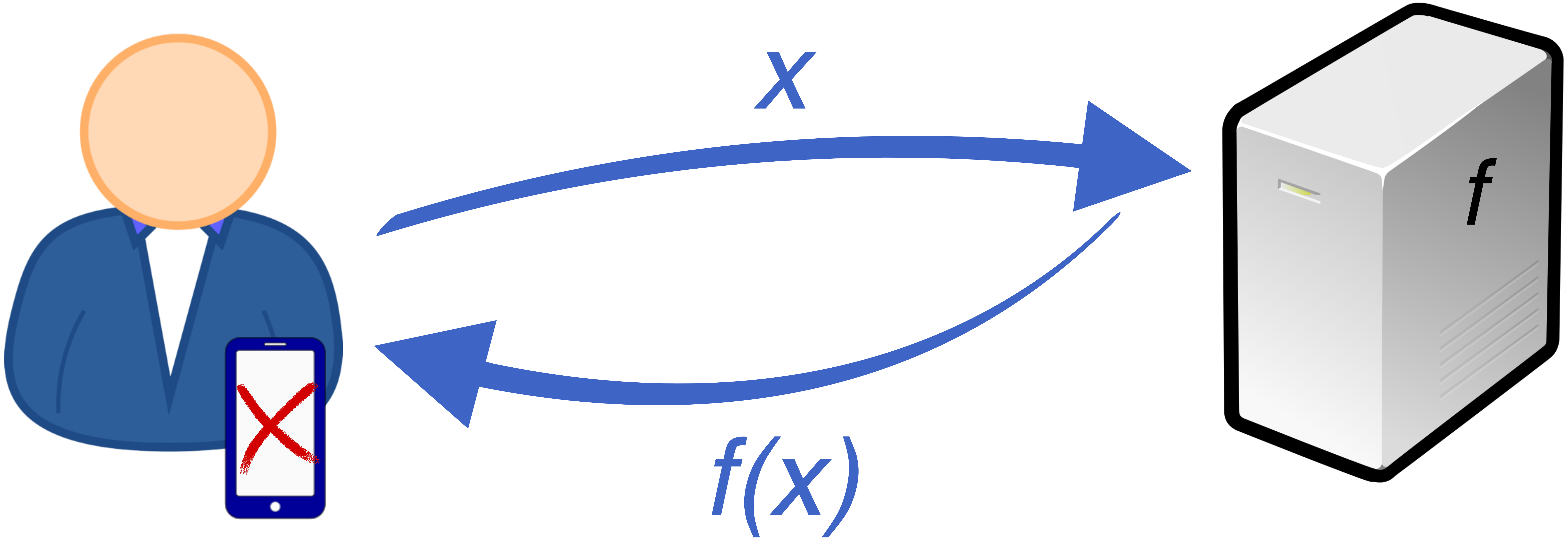
Verifiable Outsourced Computation



Verifiable Outsourced Computation

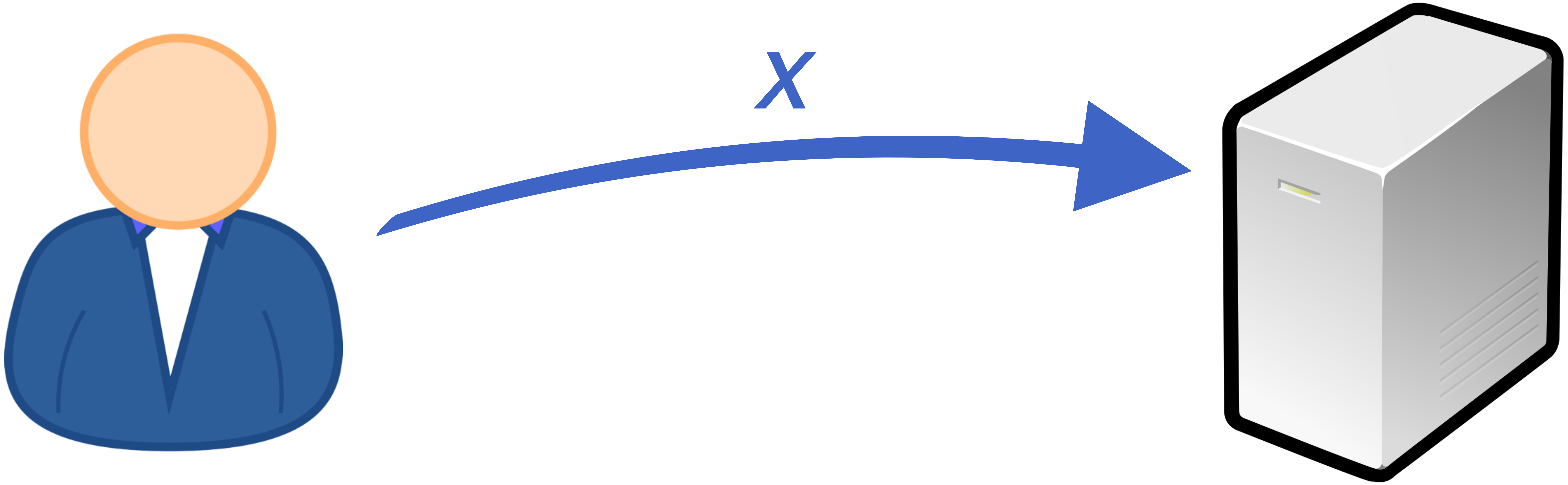


Verifiable Outsourced Computation



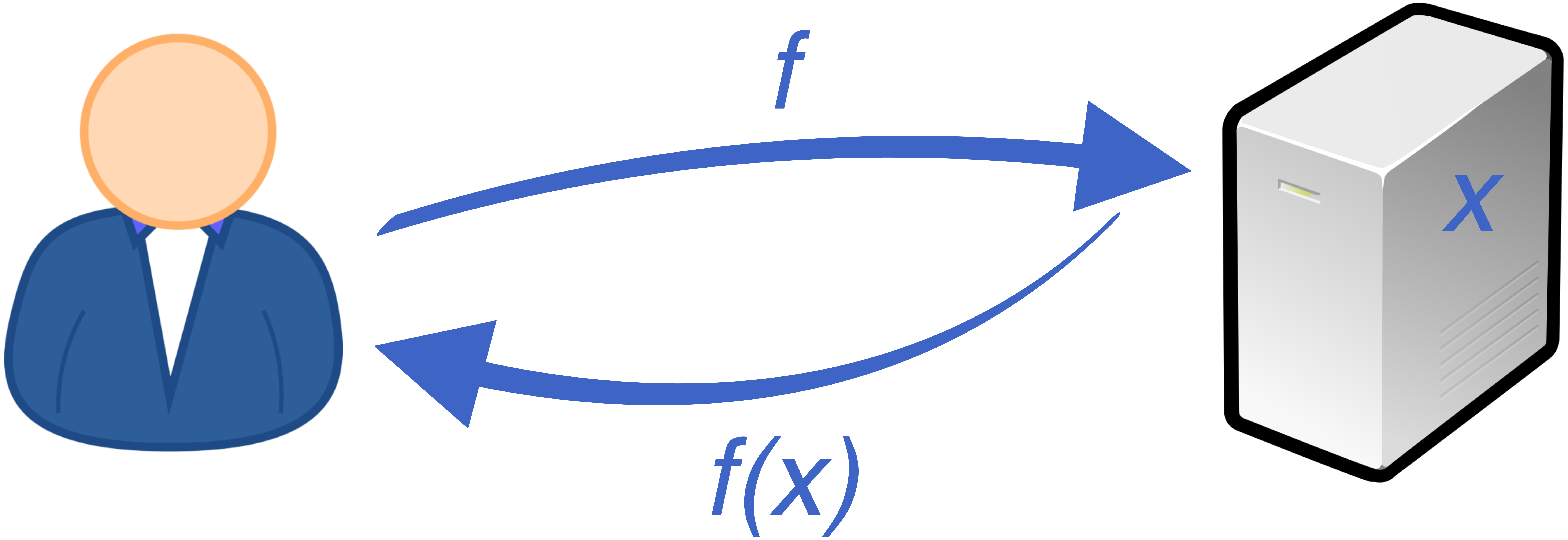
Outsourcing and verifying must be cheaper than computing f(x) locally

Searchable Encryption/Memory Delegation

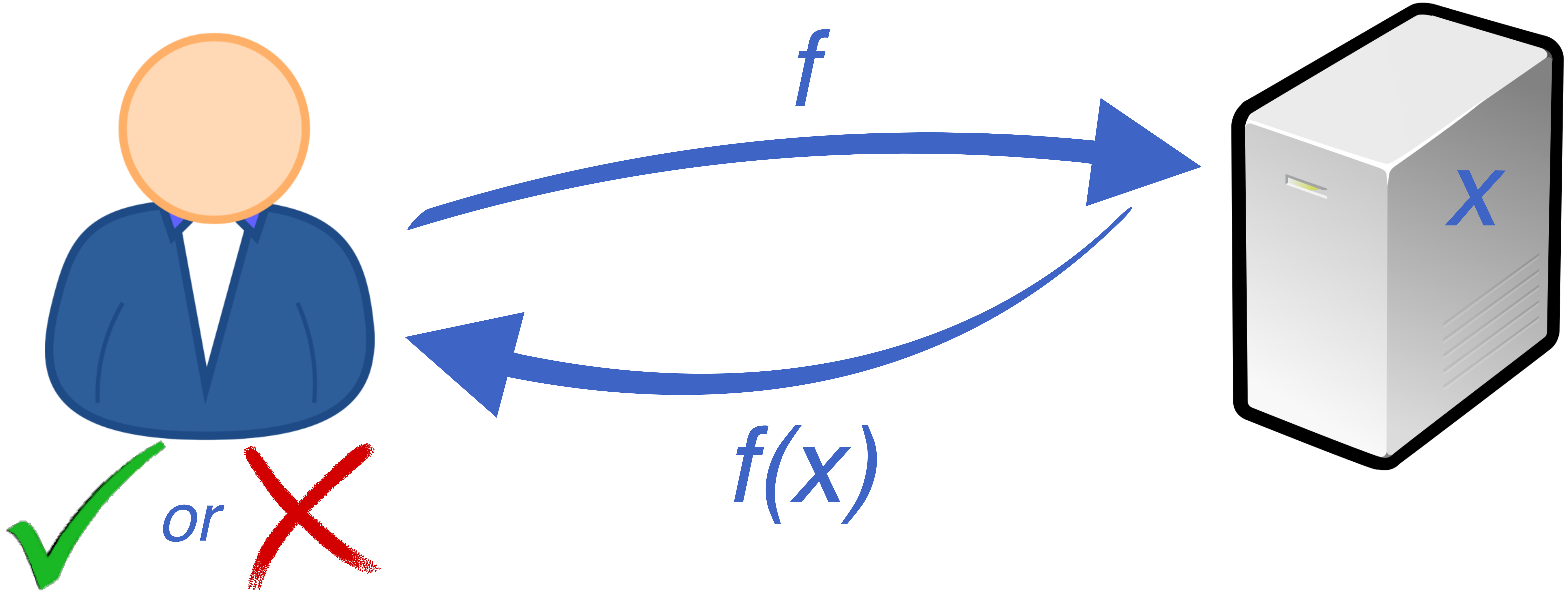


Searchable Encryption/Memory Delegation

Searchable Encryption/Memory Delegation



Searchable Encryption/Memory Delegation



ROYAL HOLLOWAY UNIVERSITY OF LONDON

Our work

Our work - a hybrid system

- Entities can act as both servers and clients as required
- Can sell spare resources to perform computations for others, or request computations when resources run low
- Data to be processed may be provided by the client or stored at the server
- Can restrict which servers can perform a given computation

Modes of Operation

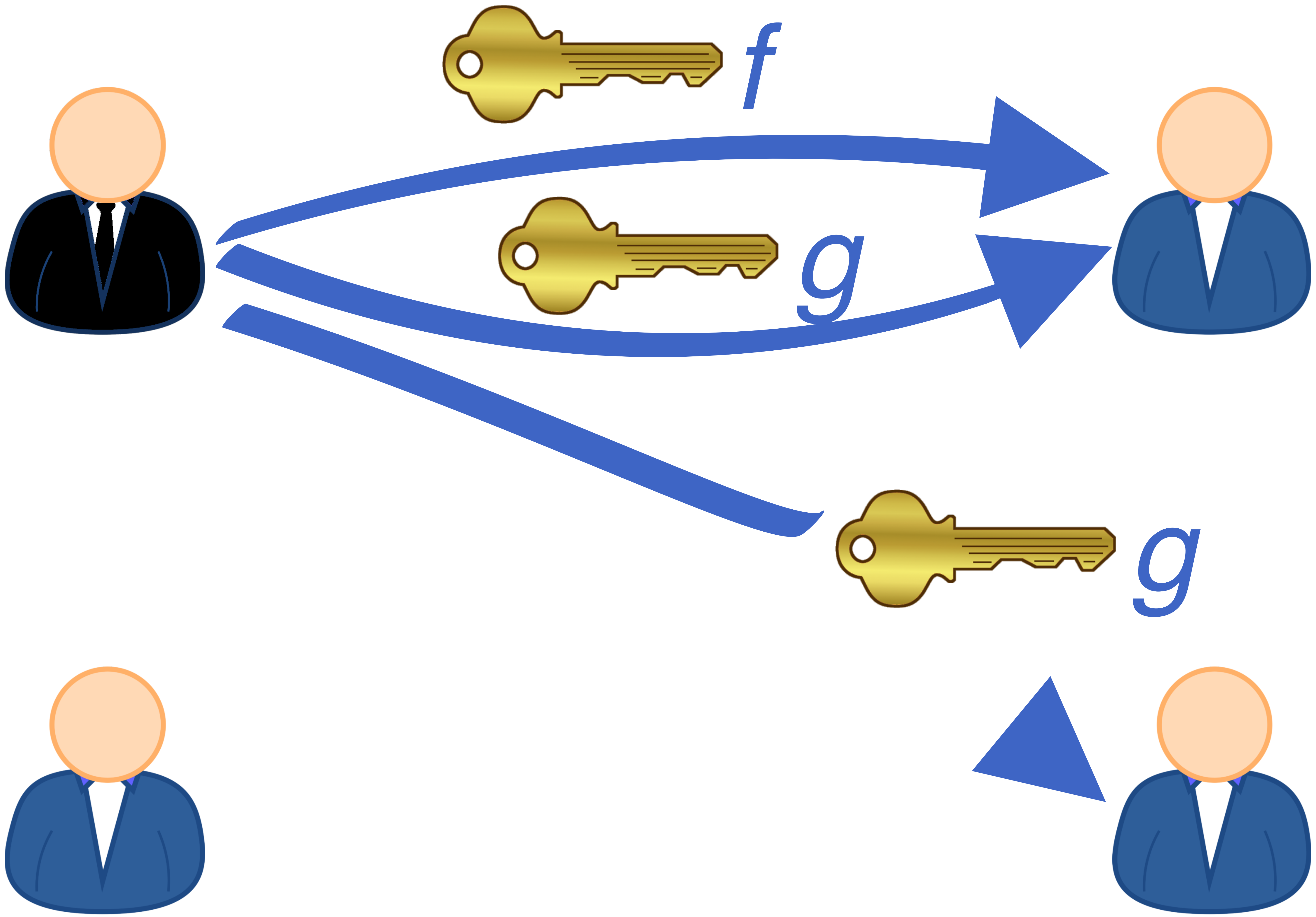
- We allow three modes of operation:
  - **Revocable Publicly Verifiable Computation (RPVC)**: client provides data, anybody can verify correctness, misbehaving servers can be revoked
  - **Revocable Publicly Verifiable Computation with access control (RPVC-AC):** as above, but can restrict the servers that may compute on a given input
  - **Verifiable Delegable Computation (VDC):** server holds data, clients request computations using public labels of the data, anybody can verify correctness

Definition

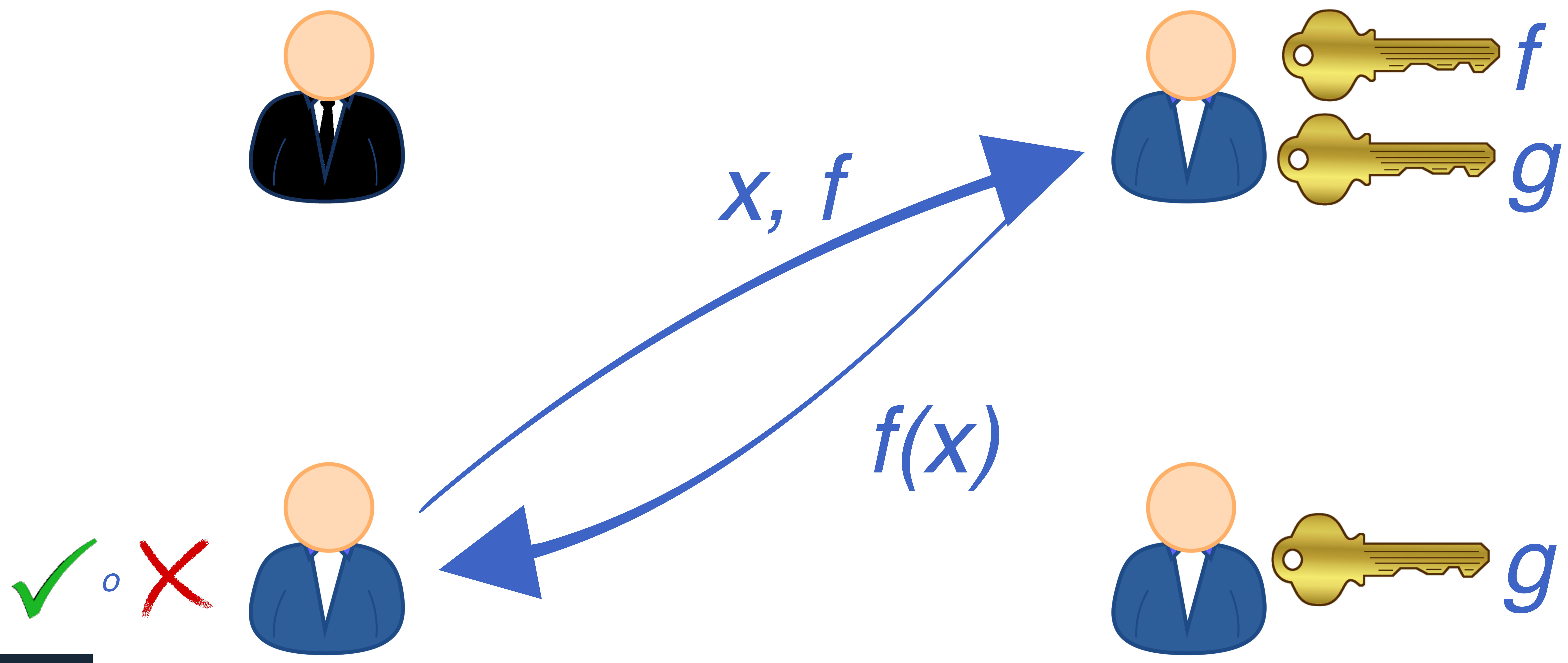
- $(PP, MK) \leftarrow$ **Setup**$(1^k, F)$
- $PK_F \leftarrow$ **FnInit**(F, MK, PP)
- $SK_S \leftarrow$ **Register**(S,MK,PP)
- $EK_{(O,\psi),S} \leftarrow$ **Certify**$(mode, S, (O, \psi), L_i, F_i, MK, PP)$
- $(\sigma_{F,X}, VK_{F,X}) \leftarrow$ **ProbGen**$(mode, (\omega, S), L_{F,X}, PK_F, PP)$
- $\theta_{F(X)} \leftarrow$ **Compute**$(mode, \sigma_{F,X}, EK_{(O,\psi),S}, SK_S, PP)$
- $(y, \tau_{F(X)}) \leftarrow$ **Verify**$(\theta_{F(X)}, VK_{F,X}, PP)$
- $UM \leftarrow$ **Revoke**$(\tau_{F(X)}, MK, PP)$

Our model

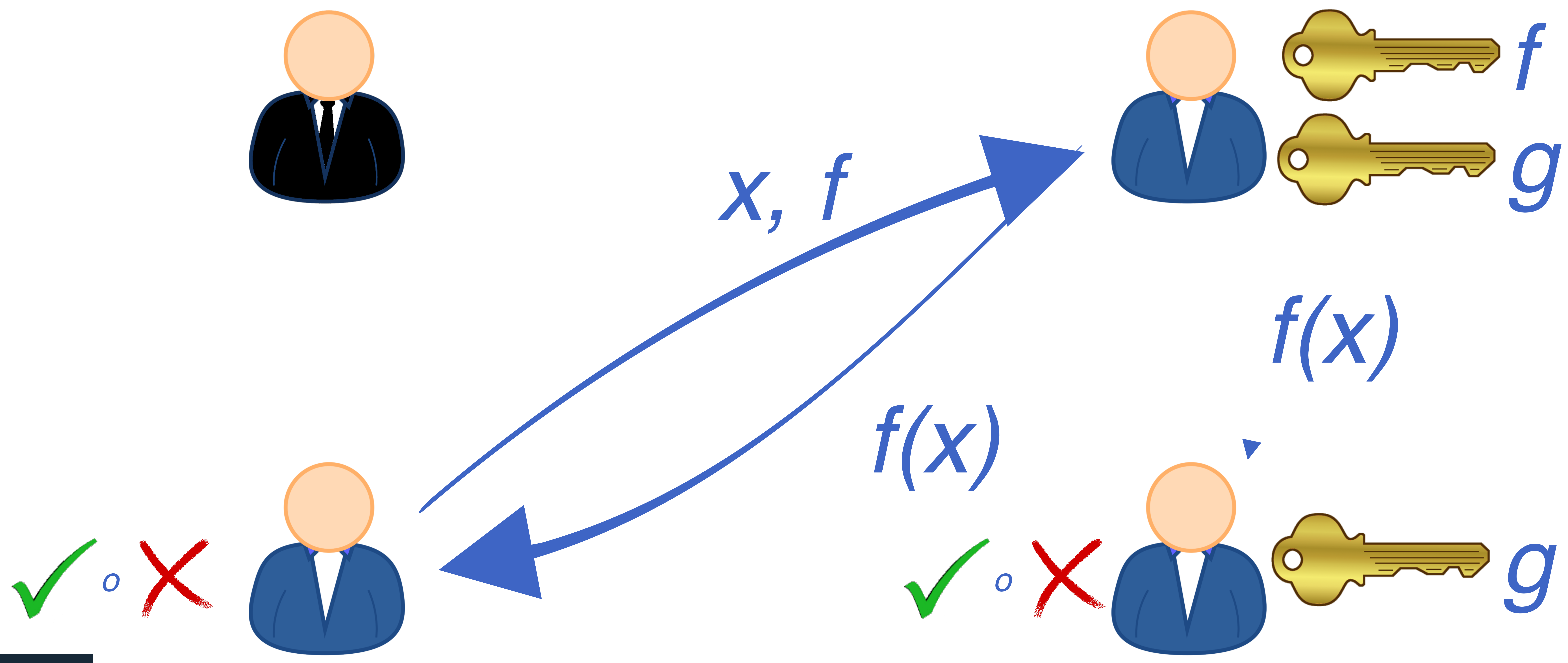
Our model - certifying servers



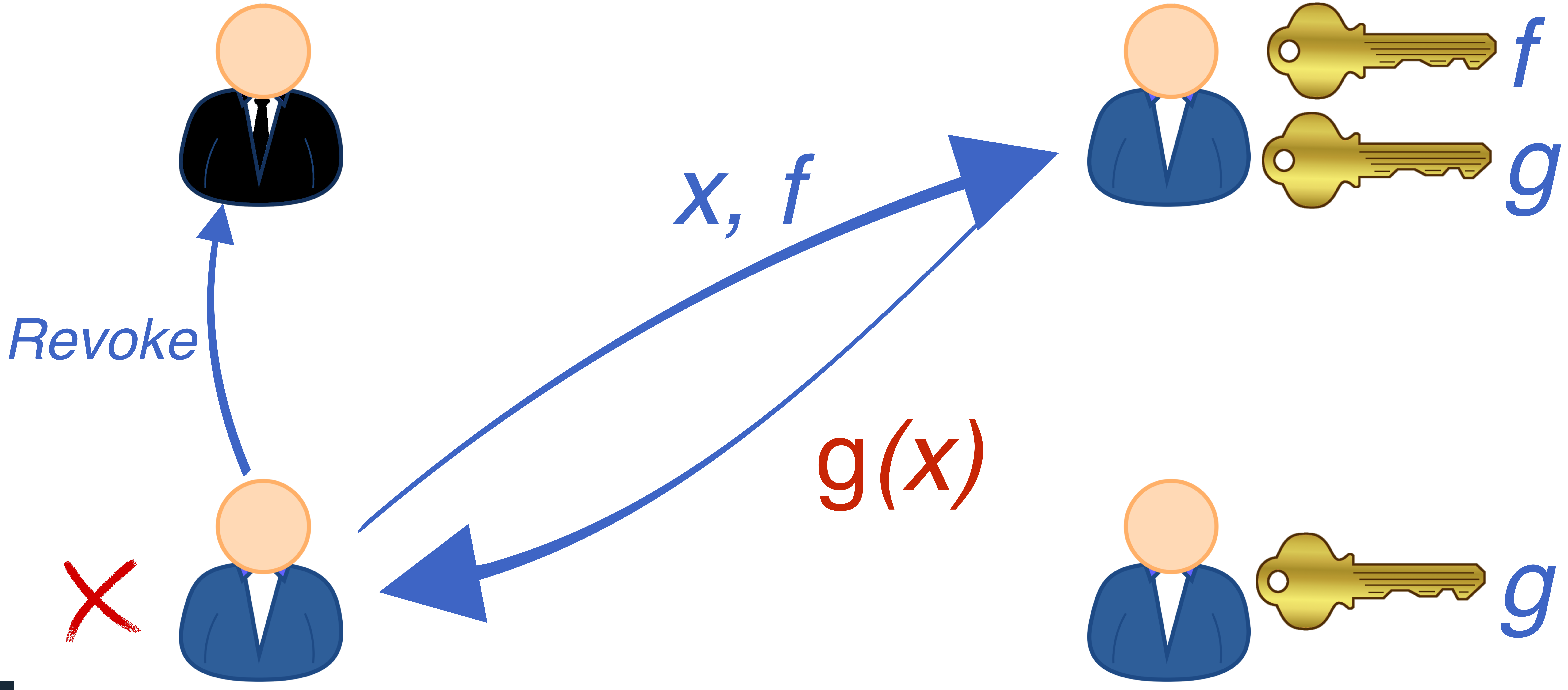
Our model - RPVC



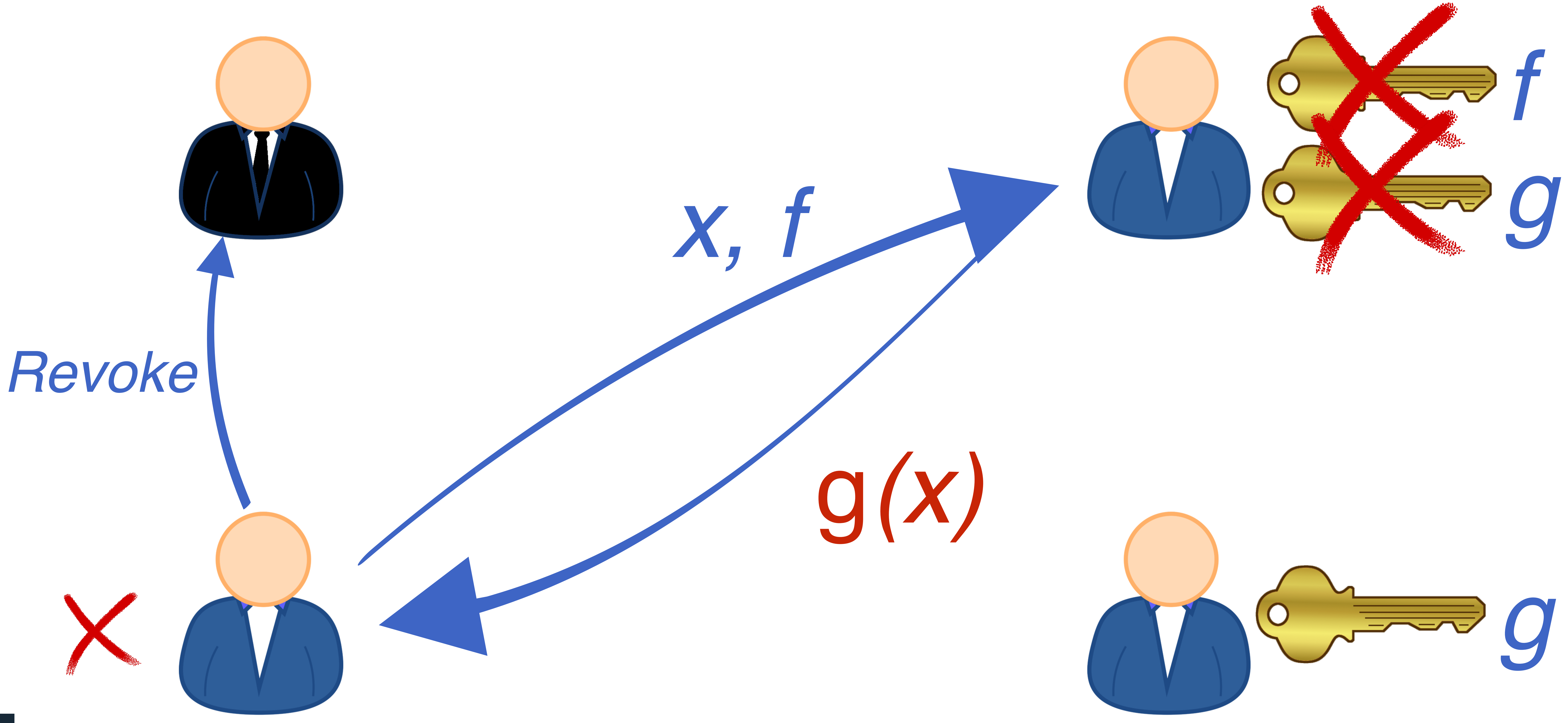
Our model - public verifiability



Our model - revocation



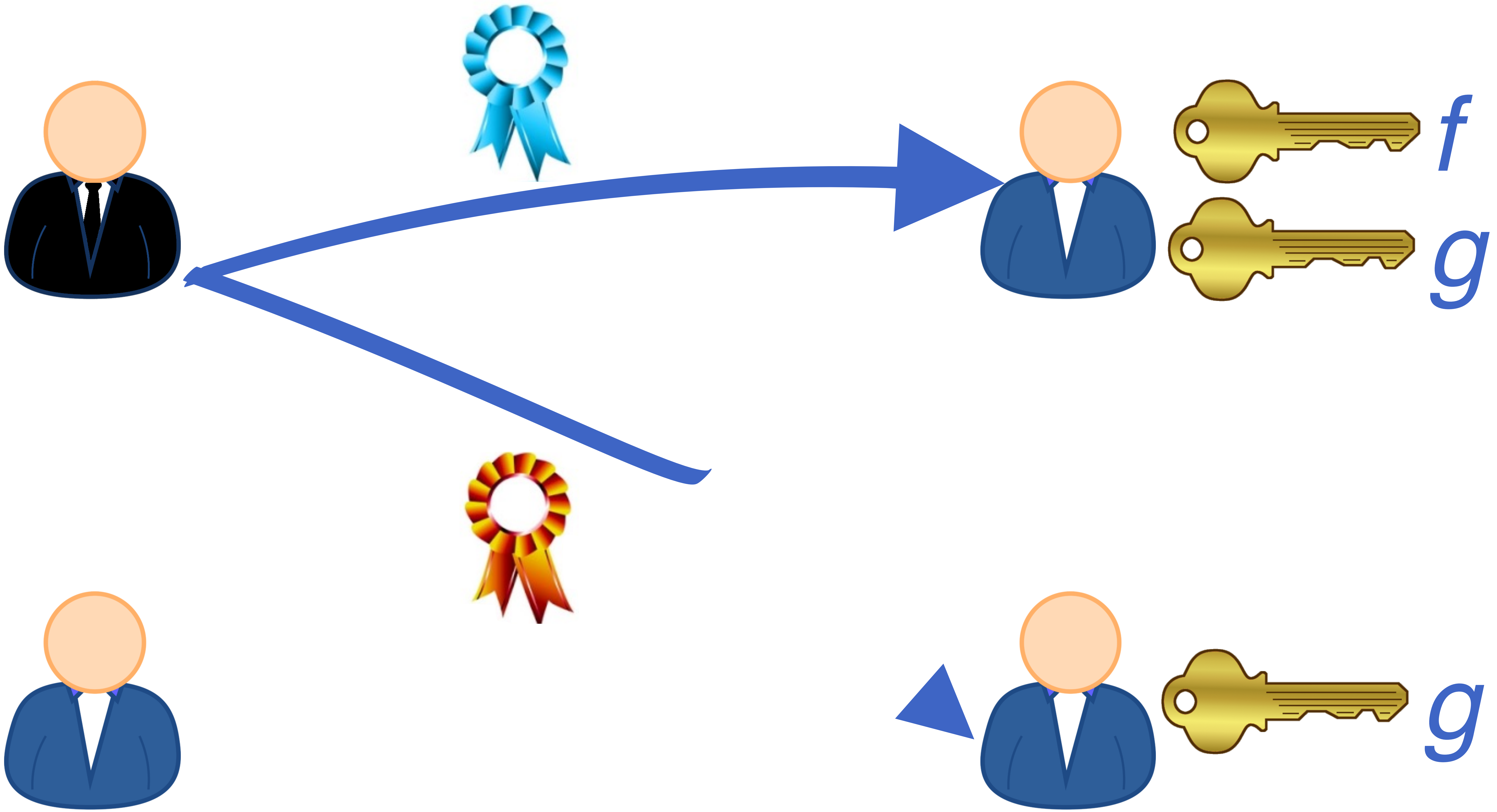
Our model - revocation



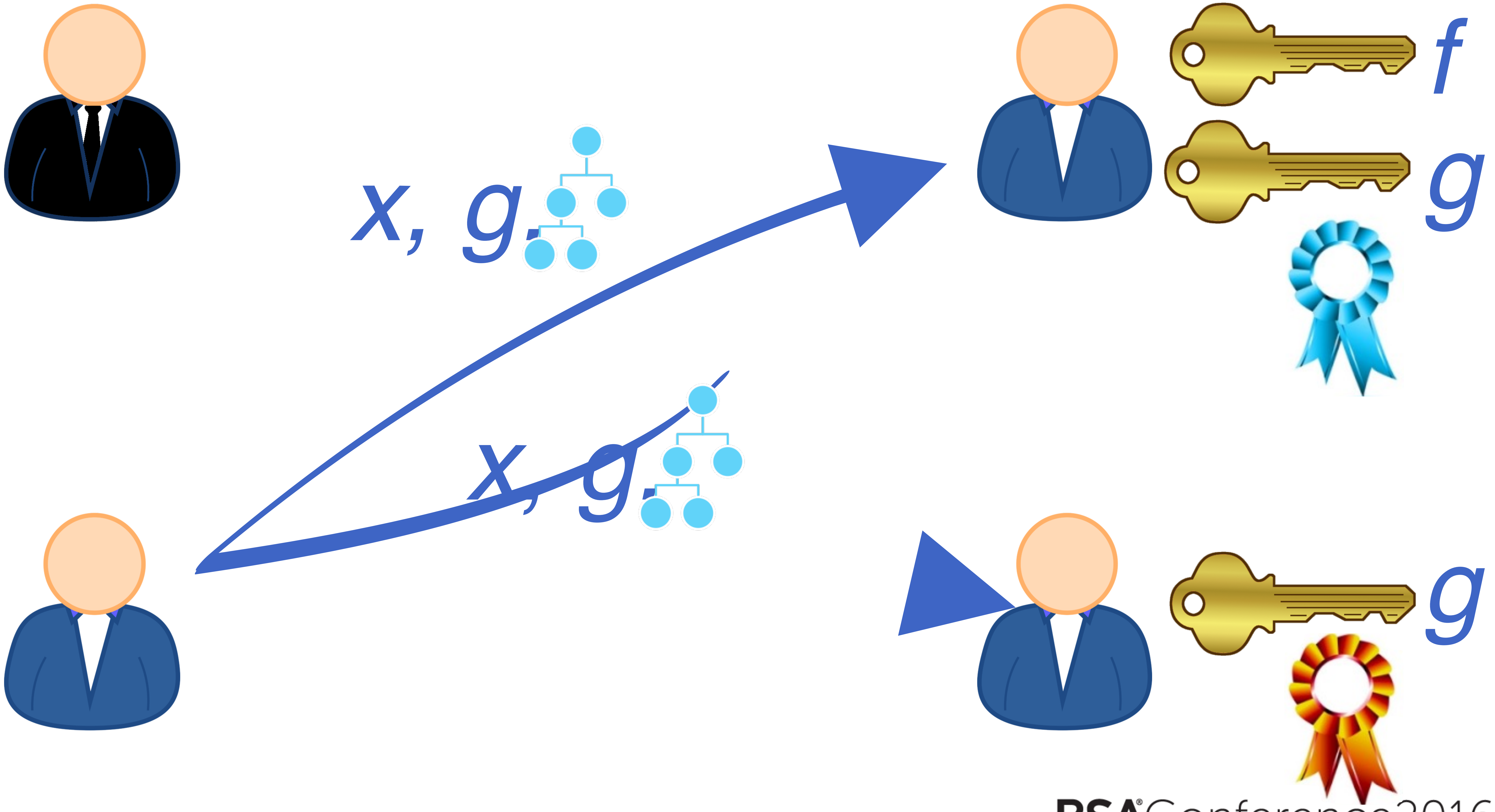
ROYAL HOLLOWAY UNIVERSITY OF LONDON

Our model - RPVC with Access Control

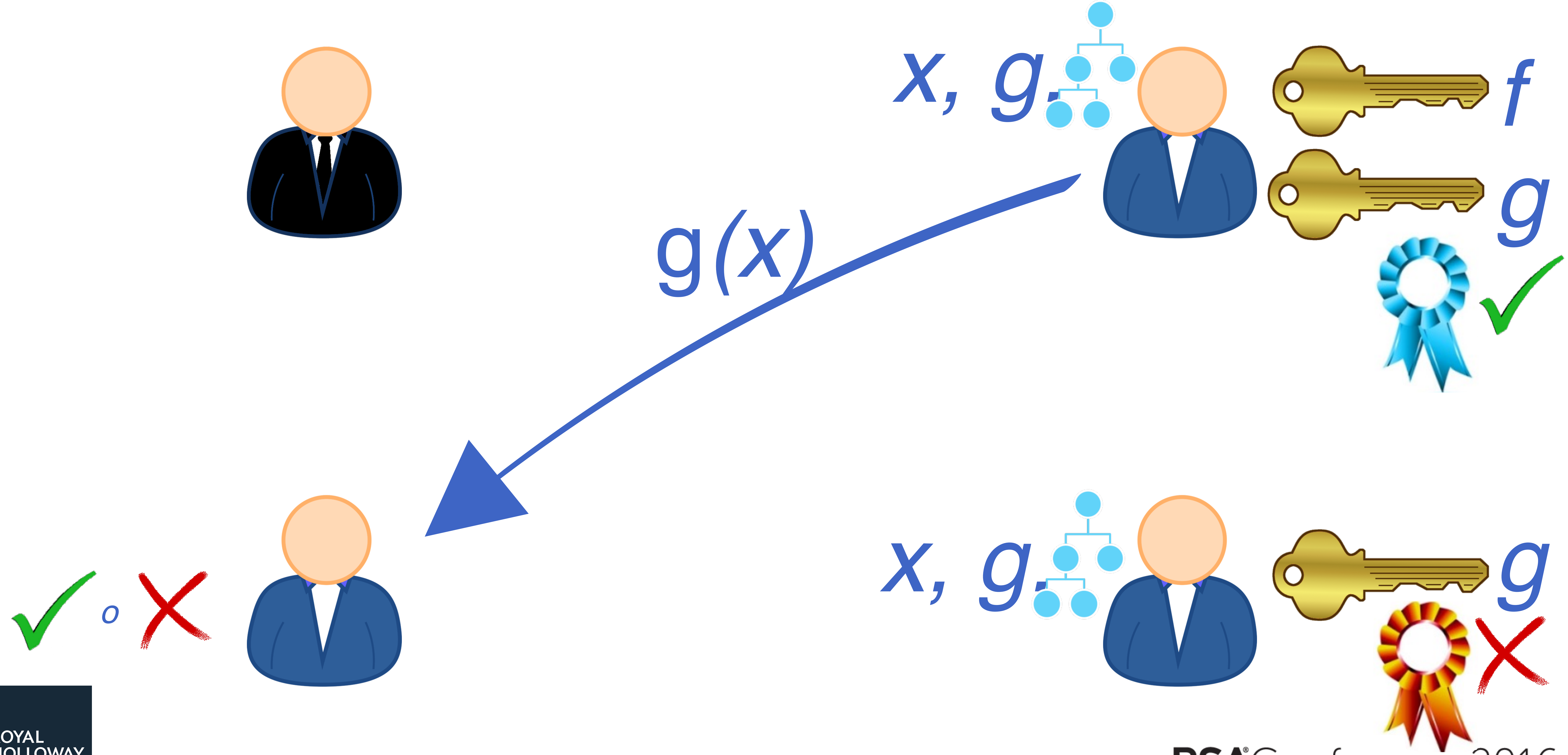
Our model - RPVC with Access Control



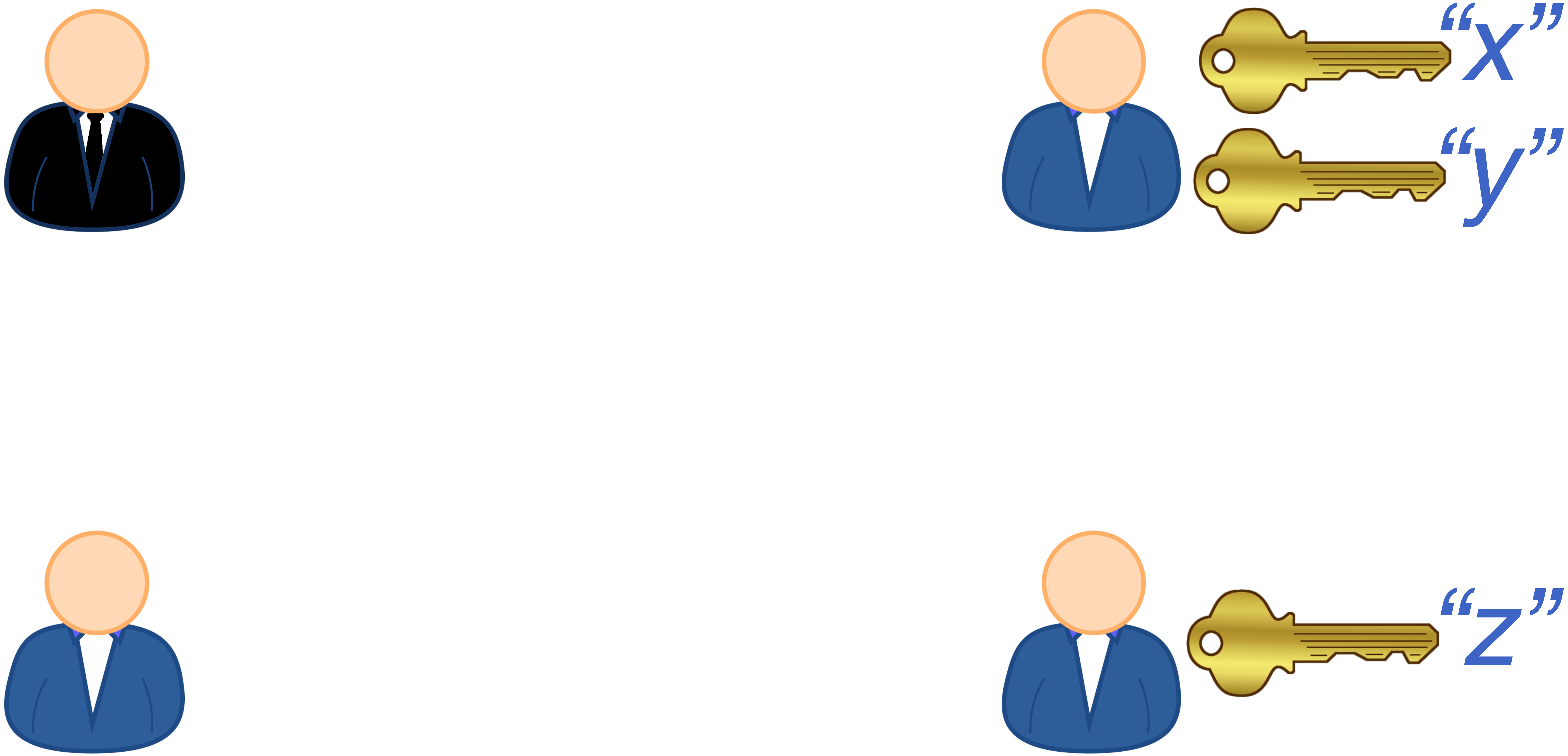
Our model - RPVC with Access Control



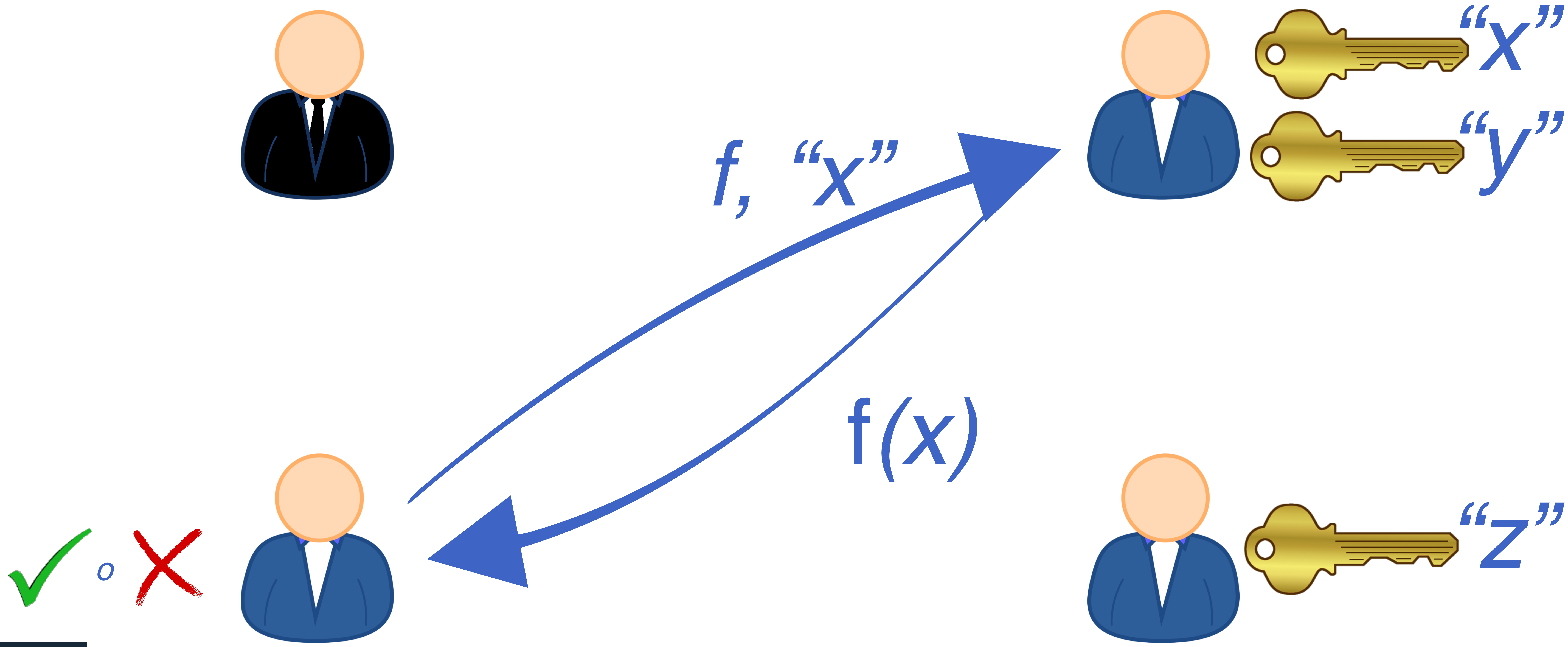
Our model - RPVC with Access Control



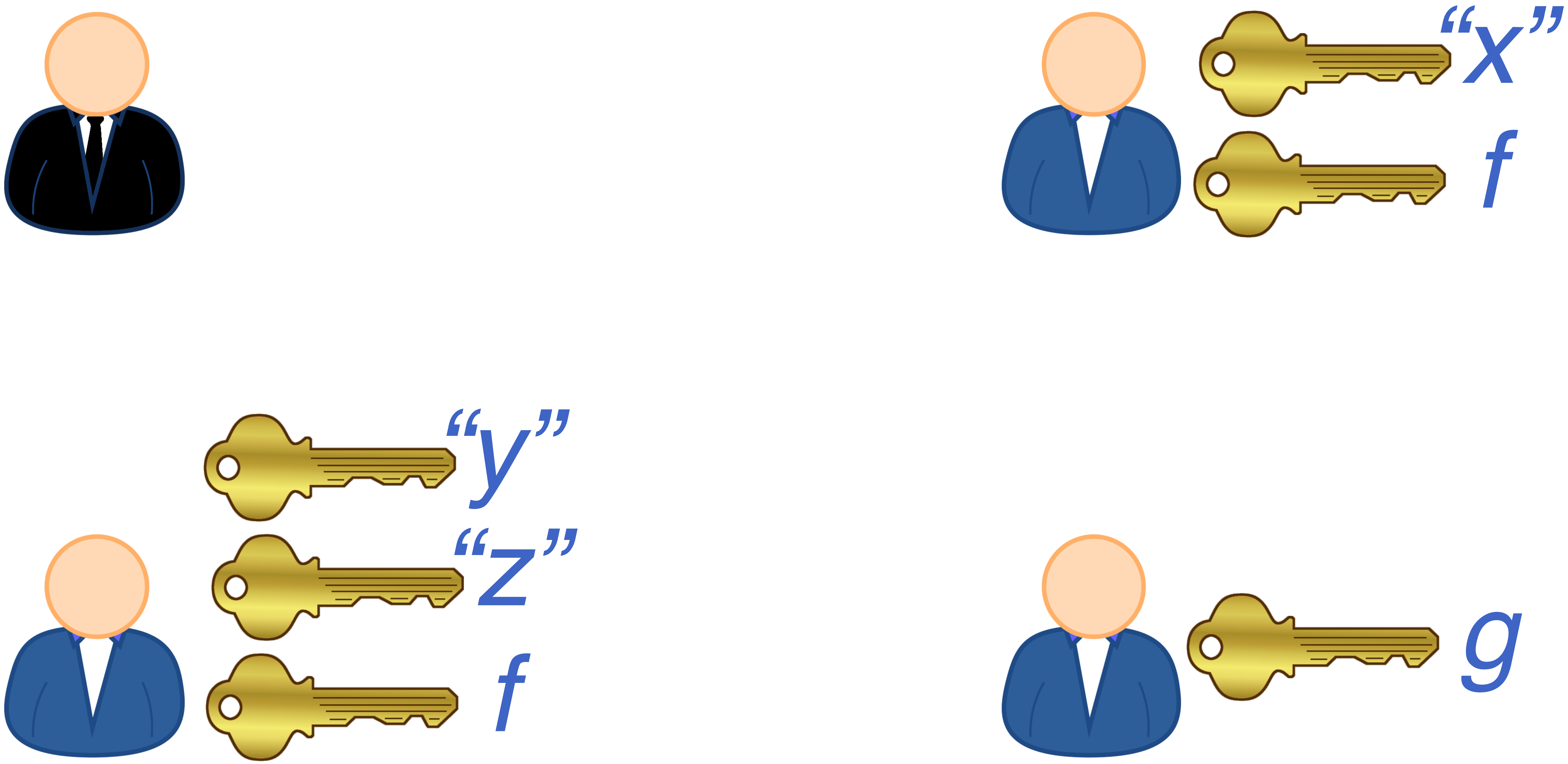
Our model - VDC



Our model - VDC



Our model - hybrid



Technical Details

Our approach

- Our approach extends the key-policy attribute-based encryption scheme of Parno et al. [TCC '12] for Boolean functions
- Functions are encoded as attribute-based policies
- Input data is encoded as attributes
- Outsourced computations are encryptions of random messages under the input attributes
- Successful decryption ⇒ Policy satisfied ⇒ Function evaluates to 1 on input. Repeat for the compliment function

Our approach

- We introduce Revocable-Key Dual-policy Attribute-based Encryption
- DP-ABE combines key-policy and ciphertext-policy attribute-based encryption
- RPVC mode uses KP-ABE (functions in server evaluation keys)
- VDC mode uses CP-ABE (data in server evaluation keys)
- RPVC with access control mode uses both — server key comprises function and authorisation attributes, ciphertext comprises input data and authorisation policy

Revocable-key DP-ABE

- (PP, MK) ← **Setup**(1^k , U)
- $CT_{(\omega,\mathcal{S}),t}$ ← **Encrypt**(m, (ω, $\mathcal{S}$), t, PP)
- $SK_{(\mathcal{O},\psi),ID}$ ← **KeyGen**(ID, ($\mathcal{O}$, ψ), MK, PP)
- $UK_{R,t}$ ← **KeyUpdate**(R, t, MK, PP)
- m ← **Decrypt**($CT_{(\omega,\mathcal{S}),t}$,($\omega$,$\mathcal{S}$),$SK_{(\mathcal{O},\psi),ID}$,($\mathcal{O}$,$\psi$),$UK_{R,t}$,PP)
  - if and only if $\omega \in \mathcal{O}$ and $\psi \in \mathcal{S}$
  - if and only if $\mathcal{O}(\omega) = 1$ and $\mathcal{S}(\psi) = 1$

$\mathcal{S}$, $\mathcal{O}$ policies
ψ, ω attribute sets

Definition

Recall:

- $(PP, MK) \leftarrow$ **Setup**$(1^k , F)$
- $PK_F \leftarrow$ **FnInit**(F, MK, PP)
- $SK_S \leftarrow$ **Register**(S,MK,PP)
- $EK_{(O,\psi),S} \leftarrow$ **Certify**$(mode, S, (O, \psi), L_i, F_i, MK, PP)$
- $(\sigma_{F, X}, VK_{F,X}) \leftarrow$ **ProbGen**$(mode, (\omega, S), L_F, X, PK_F, PP)$
- $\theta_{F(X)} \leftarrow$ **Compute**$(mode, \sigma_{F,X} , EK_{(O,\psi),S}, SK_S, PP)$
- $(y,\tau_{F(X)}) \leftarrow$ **Verify**$(\theta_{F(X)},VK_{F,X},PP)$
- $UM \leftarrow$ **Revoke**$(\tau_{F(X)},MK,PP)$

Parameter Choices

- Recall: key has policy $\mathcal{O}$ and attributes ψ
- Ciphertext has policy $\mathcal{S}$ and attributes ω

Mode	$\mathcal{O}$	ψ	ω	$\mathcal{S}$	label	F_i
RPVC	F	$\{T_o\}$	x	$\{\{T_S\}\}$	"F"	F
RPVC-AC	F	p	x	$\mathcal{P}$	"F"	F
VDC	$\{\{T_o\}\}$	x	$\{T_o\}$	F	"x"	$F_1, F_2, \ldots, F_n$

$\mathcal{S}$, $\mathcal{O}$ policies
ψ, ω attribute sets
T_o, T_S dummy attributes
p authorisation attributes
$\mathcal{P}$ authorisation policy

Security Models

- **Public Verifiability** — cheating servers are detected, servers can't use evaluation keys for different functions
- **Revocation** — revoked servers can't produce acceptable outputs
- **Authorised Computation** — only servers that satisfy the authorisation policy can produce acceptable outputs
- **Indistinguishability against selective-target with semi-static query attack (IND-sHRSS)** — security model for revocable-key DP-ABE

Summary

- We introduce a hybrid framework for flexible outsourcing of computations
  - RPVC - revocable outsourcing on local data
  - RPVC-AC - RPVC with access control policies detailing which servers can perform the computation
  - VDC - verifiable querying on remote data
- We introduce Revocable-Key Dual-policy Attribute-based Encryption to enable revocation of misbehaving entities

Thank you

eprint.iacr.org/2015/320

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Clarus
A framework for user centred privacy and security in the cloud