Towards a Theory of Trust

Jeannette M. Wing
VP, Head of Microsoft Research International
President's Professor of Computer Science, Carnegie Mellon University (on leave)

Joint work with Virgil Gligor

GREPSEC, San Francisco, CA
May 18, 2013

Motivation (inspired by Manuel Blum)

How can I (a human) trust the information I read over the Internet?

receiver

sender

Networks of Humans and Computers

Insight

- Computational trust defines trust relations among devices, computers, and networks

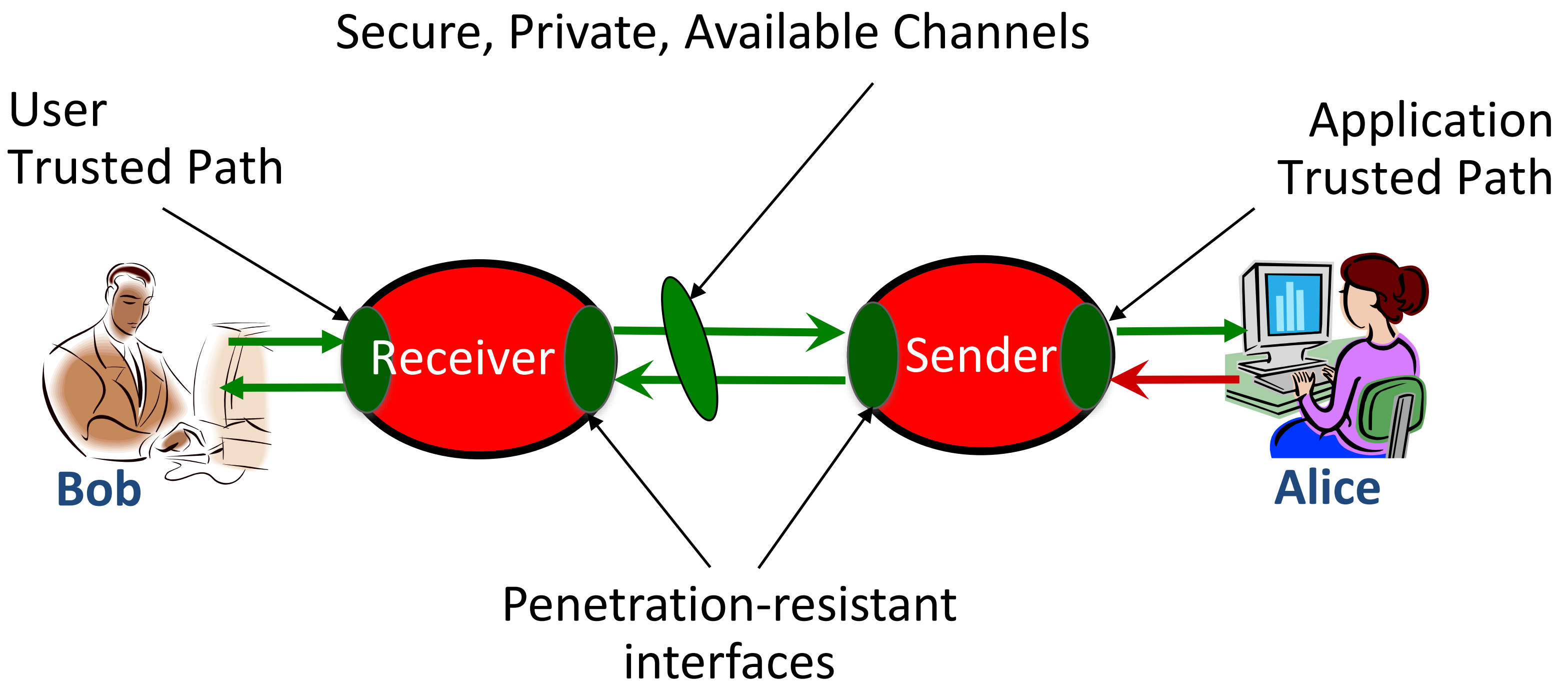
- Behavioral trust defines trust relations among people and organizations

- A theory of trust for networks of humans and computers needs to include elements of both.

Punchlines: A General Theory of Trust (for Networks of Humans and Computers)

- Needs to build on elements of computational trust and behavioral trust
  - Research (foundational): What are those elements? How do they reinforce or complement each other? How do they compose?
- Should elucidate new trust relations and show how they provide new economic value
  - Research (security economics): What are those new relations and how does one monetize them?
- Should thus suggest new computational infrastructure to support behavioral trust in a computational setting
  - Research (systems): What new computational mechanisms and systems/network architectures and protocols could support betrayal aversion?

Simple Communications Model



Decomposing Question

messages

How can I (a human) trust the information
I read over the Internet?

receive

- Is the communication channel over which I receive messages secure?
- How can I trust the sender of the messages I receive?

Decomposing Question

messag**es**

How can I (a human) trust the information
I read over the Internet?

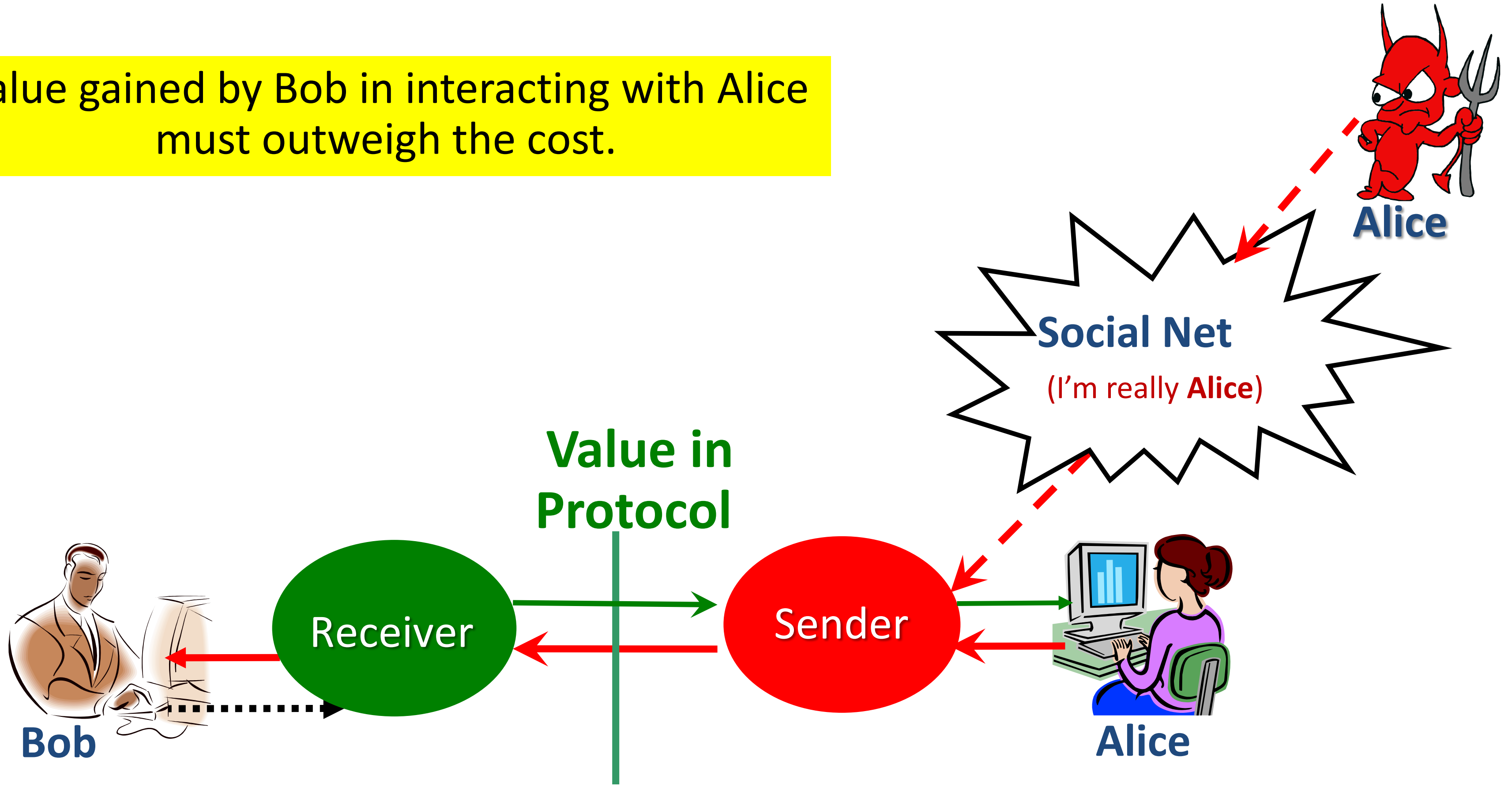
receive

- Is the communication channel over which I receive messages secure?
- **How can I trust the sender of the messages I receive?**

Our main question boils down to the *act of trusting the sender*.

Value to Receiver (Bob) in Interacting with Sender (Alice)

Value gained by Bob in interacting with Alice must outweigh the cost.



Value Underlying the *Act of Trusting the Sender*

- If Receiver trusts Sender and the Sender is trustworthy
  - Value gained (for both)
    - Receiver gets information; Sender monetizes on click
- If Receiver trusts Sender and the Sender is untrustworthy
  - Then Value gained > Cost to engage
    - Receiver risks getting malware
- If Receiver suspects Sender is untrustworthy, then don’t engage
  - Then no Value exchanged.

Computational Trust

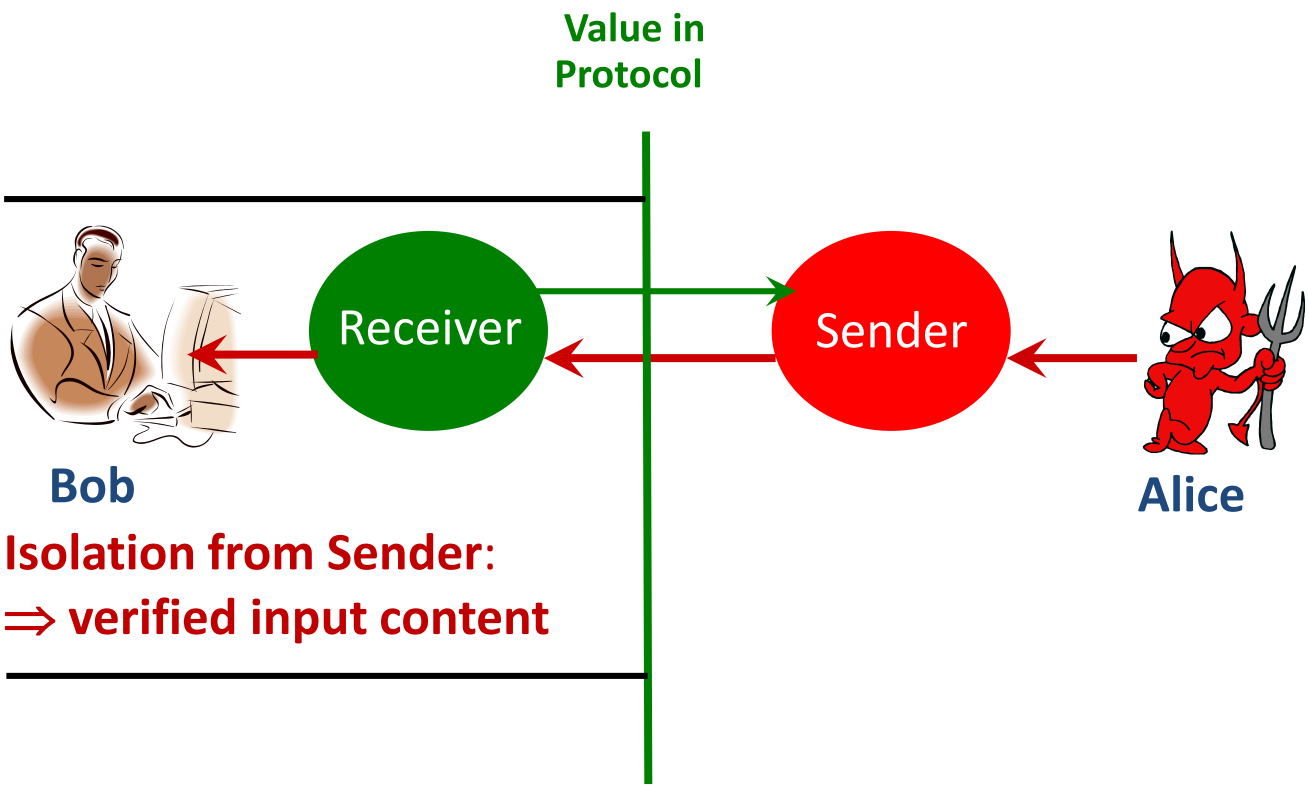
Elements of Computational Trust

- Isolation
  - Receiver could isolate himself from Sender, regardless of what/who the Sender is
- Correctness
  - Independent verification of correctness of Sender code
- Recovery
  - Detect and recover from bad input from Sender

How can I trust the sender of the messages I receive?

Necessary, but Not Sufficient

Receiver Isolation



Verification (local/outsourced, deterministic/probabilistic, etc.)

⇒ Trust in Sender is ***not*** needed

⇒ Don’t care about Alice’s behavior…

Isolation: Always Possible and Efficient?

"All trust is local" [Lampson, CACM 09]

But, can Input *always* be verified?

- *ascii*? … *pdf*? … *doc, ppt, xls*? … Java and other scripts?

No!

- Input = *arbitrary code*
- i.e., verification of code's "output behavior" by Receiver is undecidable in general

When Input can be verified, is verification *always* efficient?

No, not likely!

- Input = solution to some *co-NP complete* problem

(i.e., efficient solution at Sender & *inefficient* verification at Receiver)

Isolation: Always Practical and Scalable?

When Input verification *is efficient*, is it *always* practical?

No!

- Input = results/output of a computation outsourced to Sender

efficient result verification by Receiver [Parno 2010]

$\Rightarrow$ *fully homomorphic encryption* [Gennaro, Gentry, Parno 2010]

When Input verification *is efficient and practical*, is it *always scalable (e.g., in the Internet)*?

No!

- Input = multi-level integrity, integrity-labeled object [Biba 77]

$\Rightarrow$ *integrity-labeled closed input*

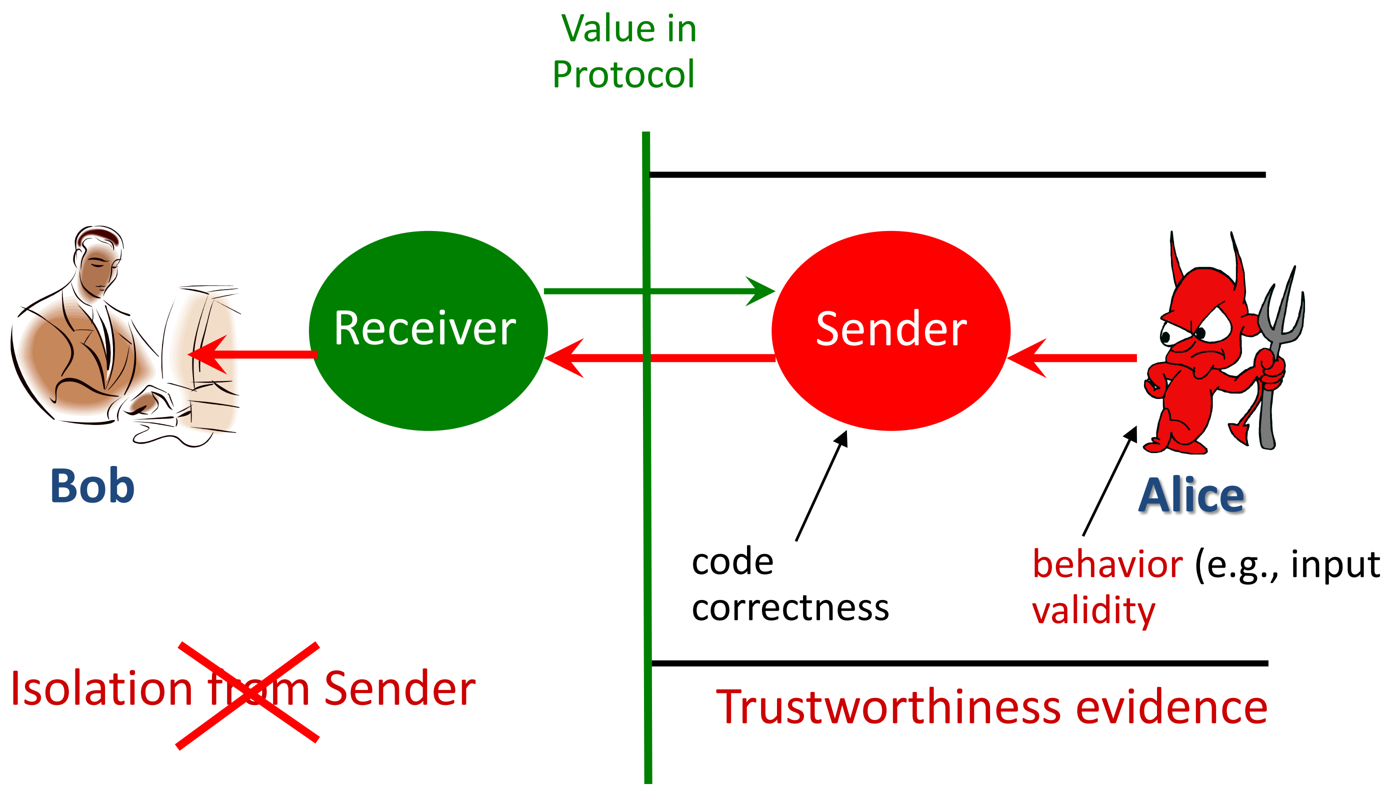
- Input = output of a trusted transaction [Clark-Wilson 87]

$\Rightarrow$ *application-closed input*

So, Receiver Isolation is Hard

Suppose Sender can provide evidence of trustworthiness?

Sender’s Trustworthiness (more than Correctness)



Sender Trustworthiness

$\Rightarrow$ No Isolation needed

$\Rightarrow$ Input is always accepted

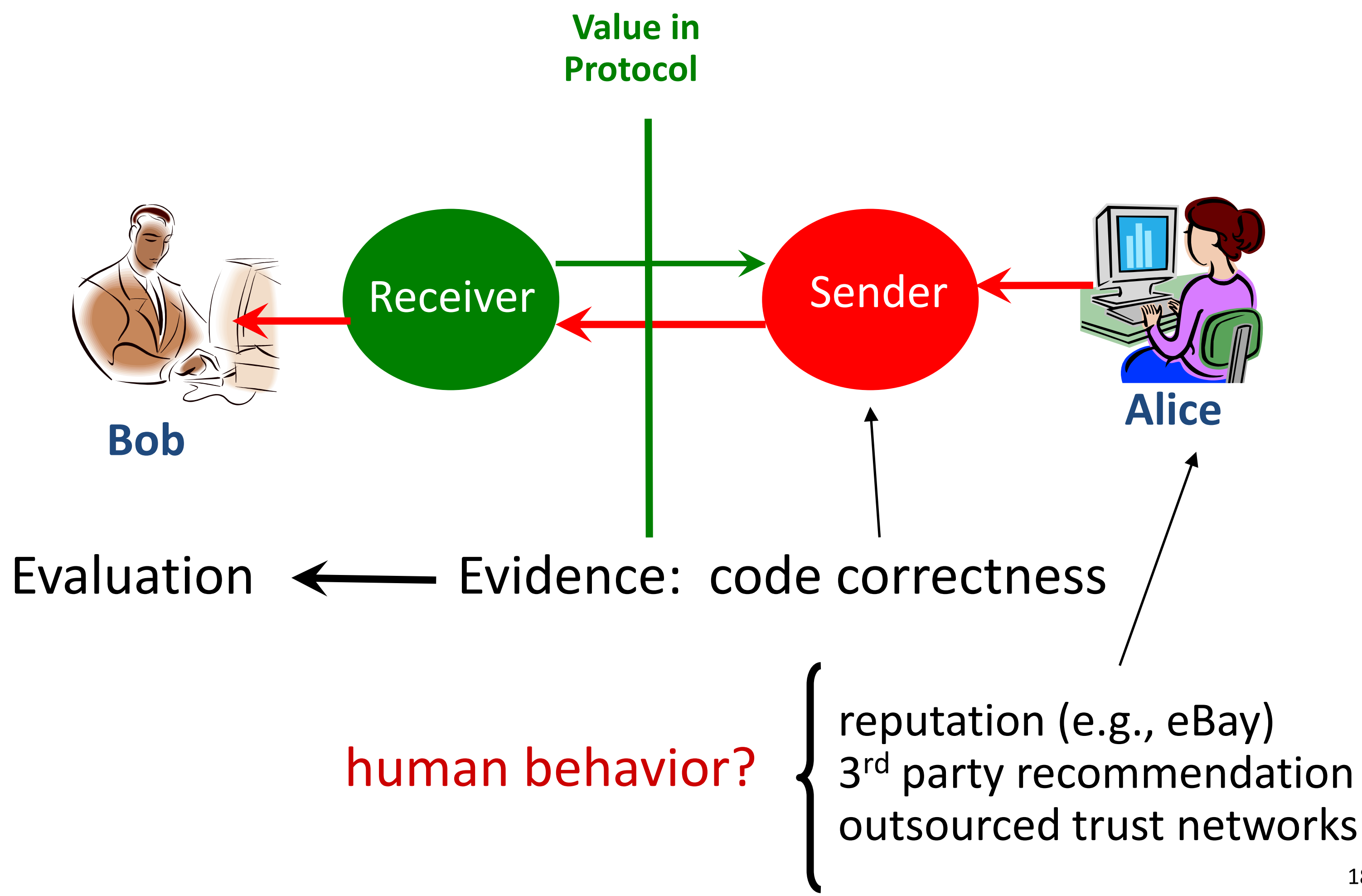
Trustworthiness Evidence: Practical?

Not usually!

- Code-correctness proofs are not “scalable”
  - limited to small configurations
    e.g., sender A is dependent on a large OS code base
    Windows, Linux, Xen (HyperVisor + root domain)
  - limited to a few properties
    e.g., configuration integrity, execution integrity
  - Assurance Approach
    e.g., TCSEC and Common Criteria Assurance levels
    - very expensive for mid- to high-level assurance
      TCSEC: B2 –> A1, CC: EAL 5 –> EAL 7

- Dependency on ***behavior*** (of many humans) for input validity

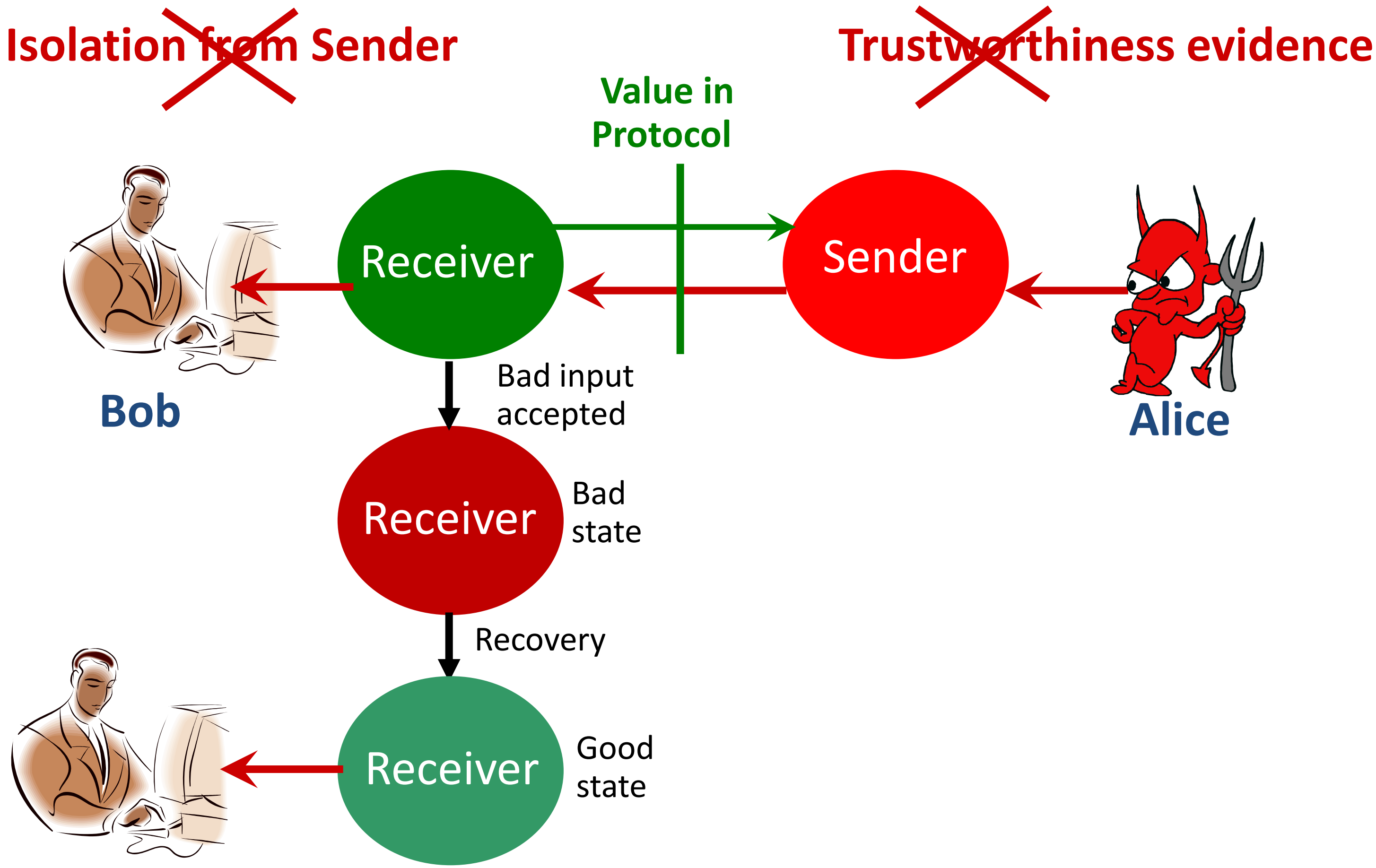
Sender's Trustworthy Behavior



So, it’s hard to provide evidence that the Sender is trustworthy.

Suppose the Receiver can detect and recover from a Sender’s untrustworthiness?

Recovery from Sender Misbehavior



Recovery ⇒ No Isolation, No Trustworthiness Needed;

⇒ Input can ***always*** be accepted

Recovery: Feasible, Practical and Scalable?

Not usually!

- Dependency on receiver state and (human input)
  - definition of state invariants
  - roll back human inputs (e.g., roll-back ingesting wrong drugs)
- It is possible in certain applications
  - transaction undo, compensation (finance, banking)
  - insurance

    Limited Assurance Approach:

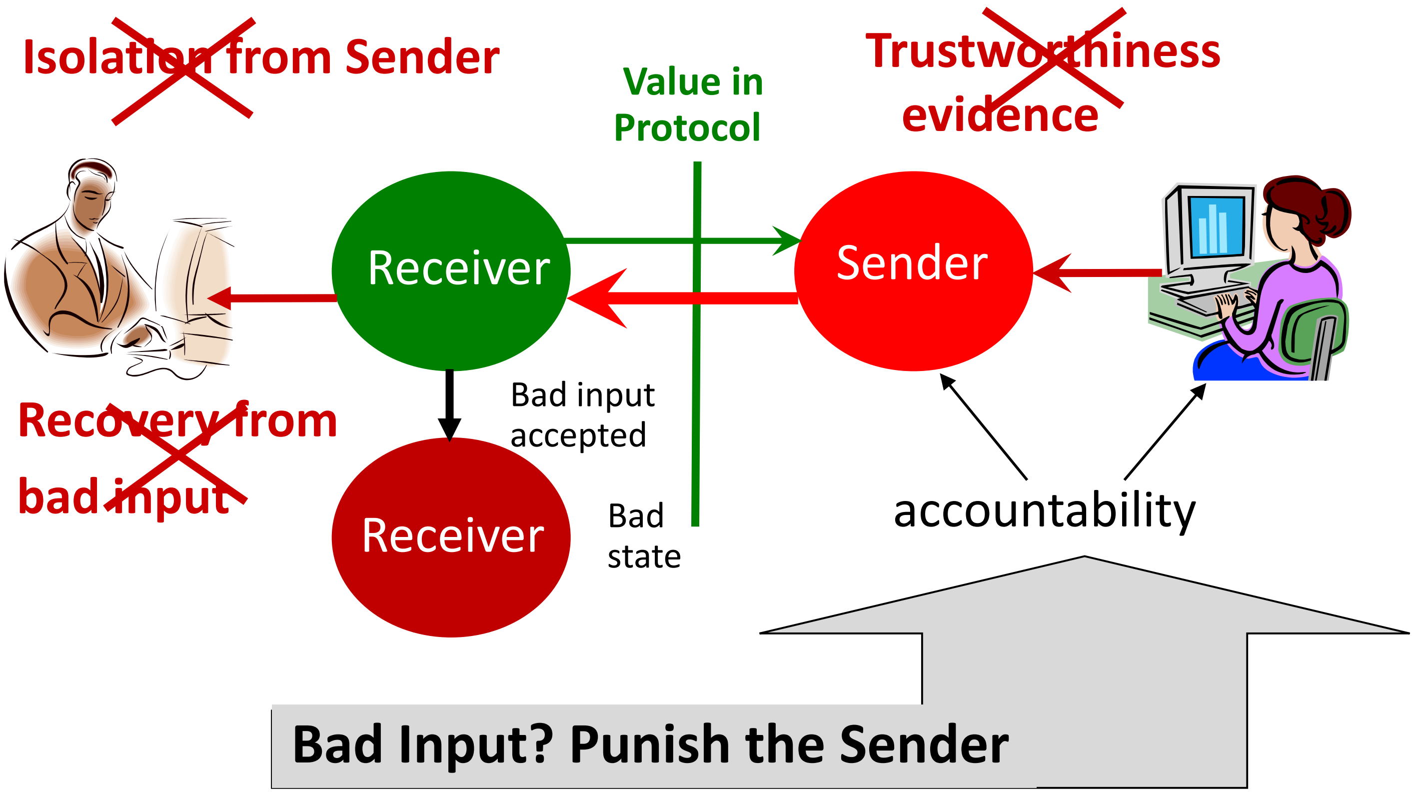
    e.g., TCSEC and Common Criteria Assurance levels

    - trusted recovery

    TCSEC: B2 –> A1, CC: EAL 5 –> EAL 7

Larger Problem: <u>Moral Hazard</u> (always, carelessly click "accept input"?)

Deter Sender (Human) Misbehavior



Deterrence $\Rightarrow$ Punishment $\Rightarrow$ Accountability [Lampson 05, CACM09]

We need $\Leftarrow$ $\Leftarrow$

sufficient **punishment to deter** and

sufficient **accountability to punish**

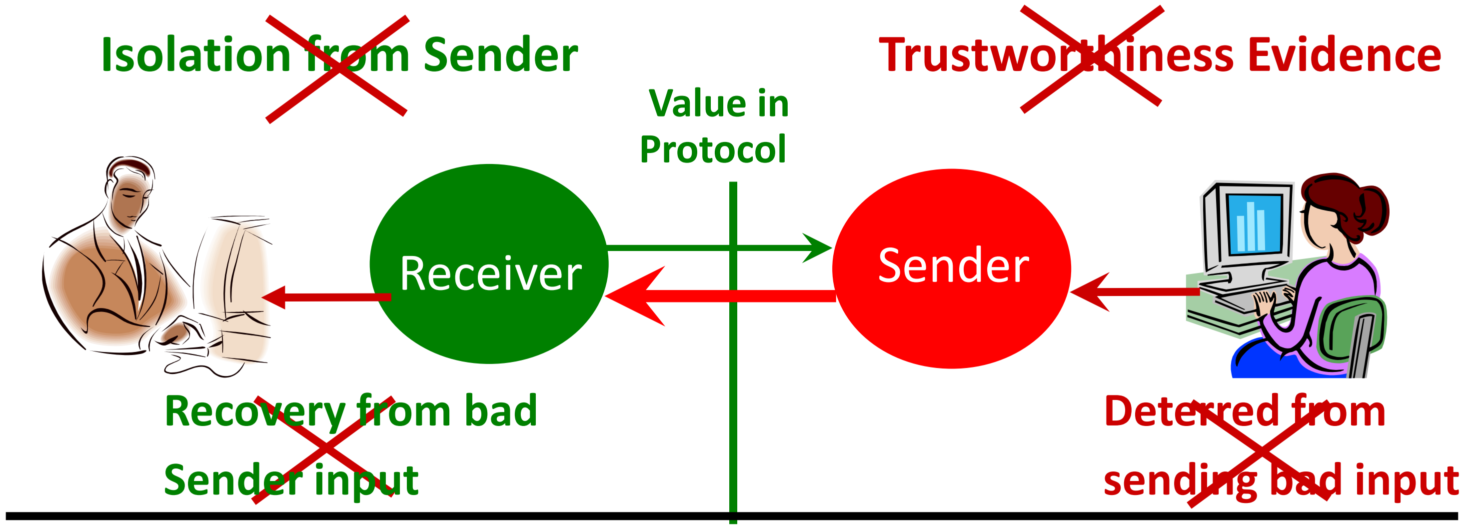
Deterrence: Always Practical, Scalable?

No, not always!

- What deters human *misbehavior*? (legal debate for centuries)

- Social norms, contract enforcement, law
    - some empirical evidence that Social Accountability deters more than the Law [CACM 2011]
    - norms-based punishment [Akerlof 2010]

The Act of Trusting



If 0% Isolation and 0%Trustworthiness Evidence and 0% Recovery and 0% Deterrence,

then the Sender is Trusted 100% . . .

and welcome to the <u>Internet of today!</u>

Is it (ever) Safe to Trust the Sender?

Theory of Trust, So Far

A theory of trust builds on these computational trust mechanisms

- Cryptography
- Verification
- Fault-tolerance

but we need more, to define trust among humans.

Behavioral Trust

The Act of Trusting

What could the *act of trusting* mean?

- Examples/theories of trust in Economics, Sociology, Psychology …

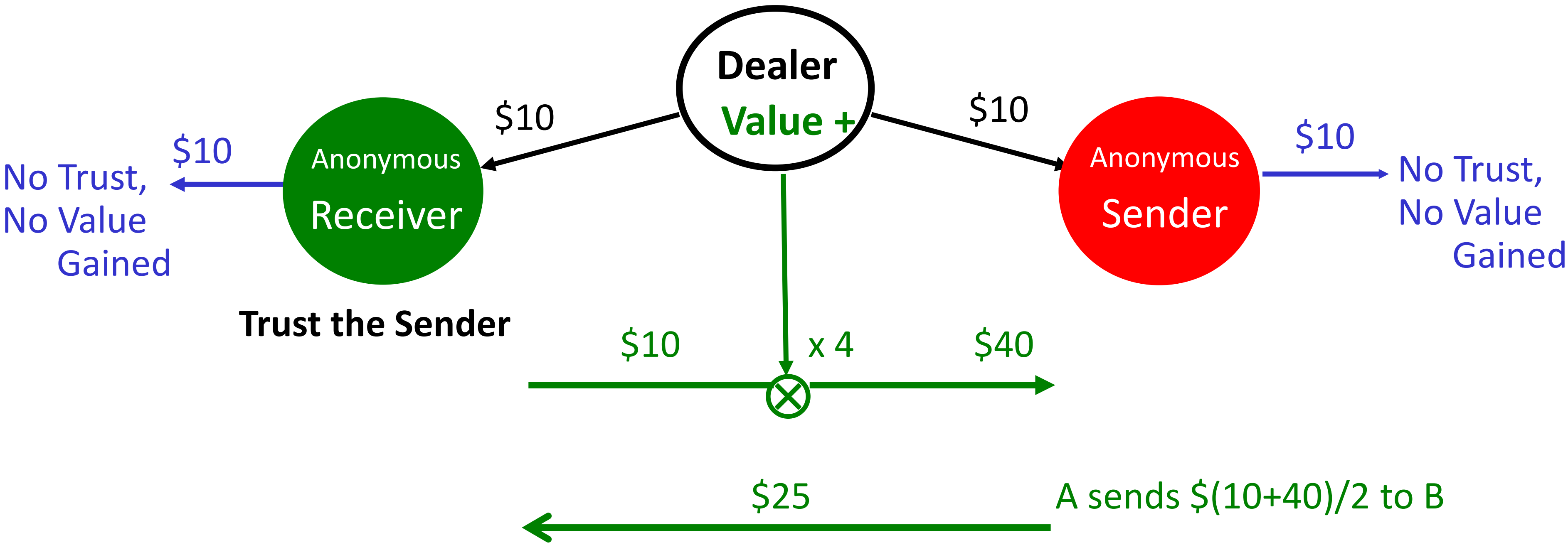
  … 100’s of research articles published to date

- Behavioral Trust [Fehr09]
  - ***beliefs*** and ***preferences*** (and nothing else)
  - commonality with computer security
  - explains role of *Deterrence, Trustworthiness, Recovery too*

A Model for Behavioral Trust

- Sender is Trustee
  - e.g,. Bank, eBay, Google, Amazon
- Receiver is Trustor (aka Investor)
  - e.g., bidder, customer
- One-Shot Game

One-Shot Trust Game



B gets $25, gaining $15

Cooperation: Win-Win

A keeps $25, gaining $15

B gets $0, losing $10

Non-compliance: Loss--Win

A keeps $50, gaining $40

B loses

A cheats

$25 - $10 = Value of Trusting Player 2

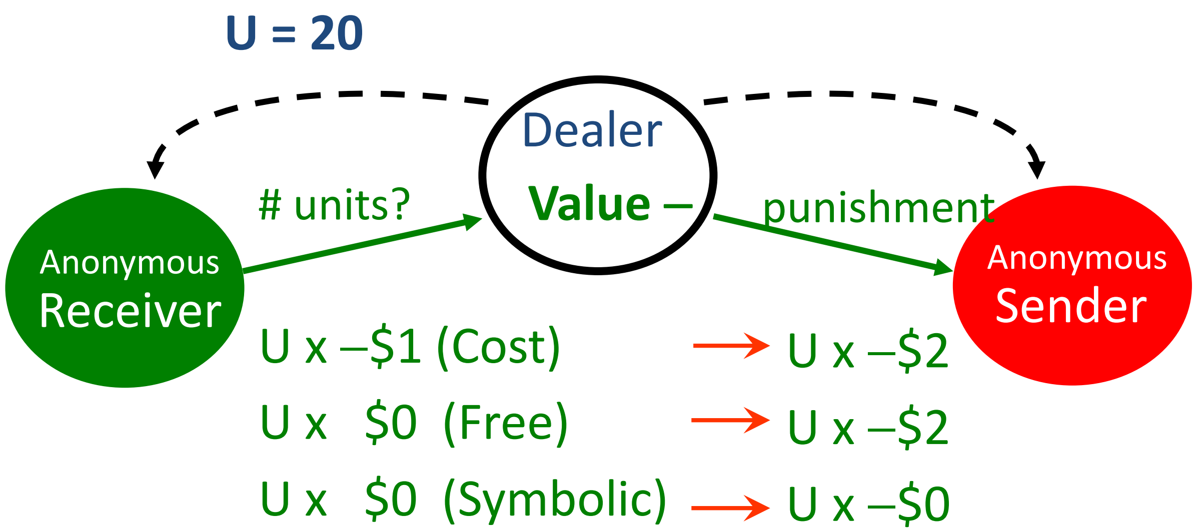
Possible Value Outcomes

Analogous to Sender-Receiver Interaction in Networks

- If trustor trusts trustee and the trustee is trustworthy
  - Then trustor and trustee are better off before executing protocol, i.e., cooperation pays off
- If trustor trusts trustee and the trustee is untrustworthy
  - Then trustee is better off and trustor is worse off, i.e., trustee has strong incentive to cheat in the absence of a mechanism that protects the trustor
- If trustor suspects trustee will cheat, then don't engage, i.e., no value exchanged.

- If Receiver trusts Sender and the Sender is trustworthy
  - Value gained (for both)
    - Receiver gets information; Sender monetizes on click
- If Receiver trusts Sender and the Sender is untrustworthy
  - Then Value gained > Cost to engage
    - Receiver risks getting malware
- If Receiver suspects Sender is untrustworthy, then don't engage
  - Then no Value exchanged.

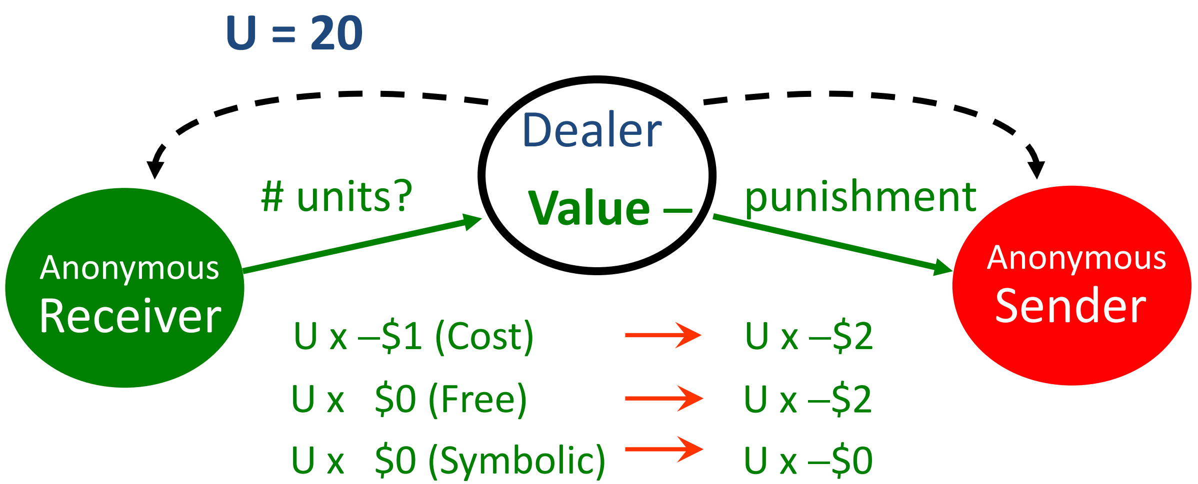
Punishment . . . [de Quervain et al. 04]



Punishment: Most **Receivers** paid **Dealer** to punish cheating **Senders**

(12/14)	**Cost ~ 11 U**	punishment:	**~ – $22**
(14/14)	**Free ~ 18 U**		**~ – $36**
(3/14)	Symbolic		

Betrayal Aversion



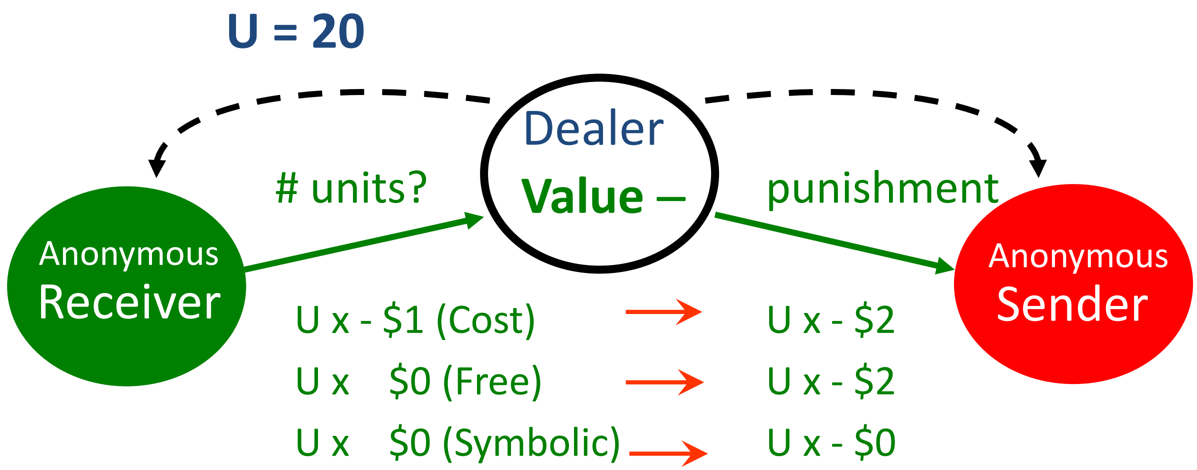
Punishment: Most **Receivers** paid **Dealer** to punish cheating **Senders**

Cost ~ 11 U (1 U → –$1 cost) punishment: **~ –$ 22**
Free ~ 18 U (1 U → $0 cost) **~ –$ 36**

PET scan of **Receiver's** brain striatum shows **reward satisfaction**

- **betrayal aversion** (e.g., aversion to being scammed, cheated)
- (biological not psychological) **altruistic punishment**

Betrayal Aversion ≠ Risk Aversion



Punishment: Most **Receivers** paid **Dealer** to punish cheating **Senders**

Cost ~ 11 U (1 U → –$1 cost**) punishment: **~ –$ 22
Free ~ 18 U (1 U → $0 cost) ~ –$ 36

PET scan of **Receiver's** brain striatum shows **reward satisfaction**
- **betrayal aversion** (e.g., aversion to being scammed, cheated)
- (biological not psychological) **altruistic punishment**

1) Betrayal Aversion ≠ Risk Aversion: Sender is a *random process*

⇒ **Receiver**: **no** (small desire) to punish and **no** (little reward) satisfaction

cost ~ 2U **punishment: < $4**

2) Oxytocin affects betrayal, but *not* risk aversion, *nor* trustworthiness beliefs

Summary of Experiment's Results

1. Trustor/Receiver is willing to incur a cost to punish, and the amount of punishment inflicted was higher when the punishment was free.

2. Trustor/Receiver derived satisfaction (i.e., felt rewarded) proportional to the amount of punishment inflicted on cheating Trustee/Sender.
   – That is, the stronger the satisfaction Trustor/Receiver derived, the higher the cost he was willing to incur. This indicates the strength of B's aversion to being betrayed by A. It also illustrates the fact that B's punishment is altruistic, since he is willing to pay to punish even though he is not deriving any material gain.

3. When the Trustee/Sender is replaced by a random device, Trustor/Receiver's desire to punish is negligible.
   – This indicates that B's aversion to the risk of losing money when faced with an ambiguous outcome was different (i.e., lower) from his aversion to being betrayed.

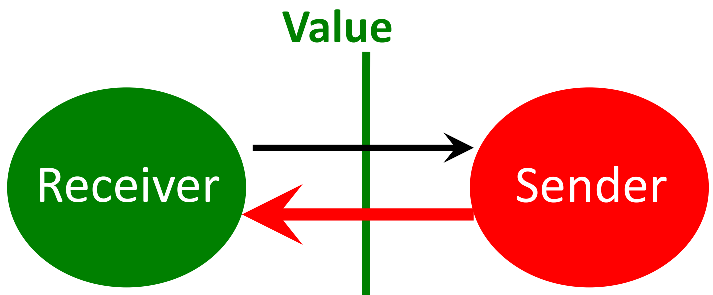
Elements of Behavioral Trust: Preferences and Beliefs

- Trustor's beliefs in trustworthiness of trustee
  - Probabilistic beliefs about a trustee's actions
- Trustor's risk preferences
  - Degree of risk aversion
- Trustor's social preferences
  - Degree of betrayal aversion

Beliefs

Preferences

Behavioral Trust Primitives from Economics



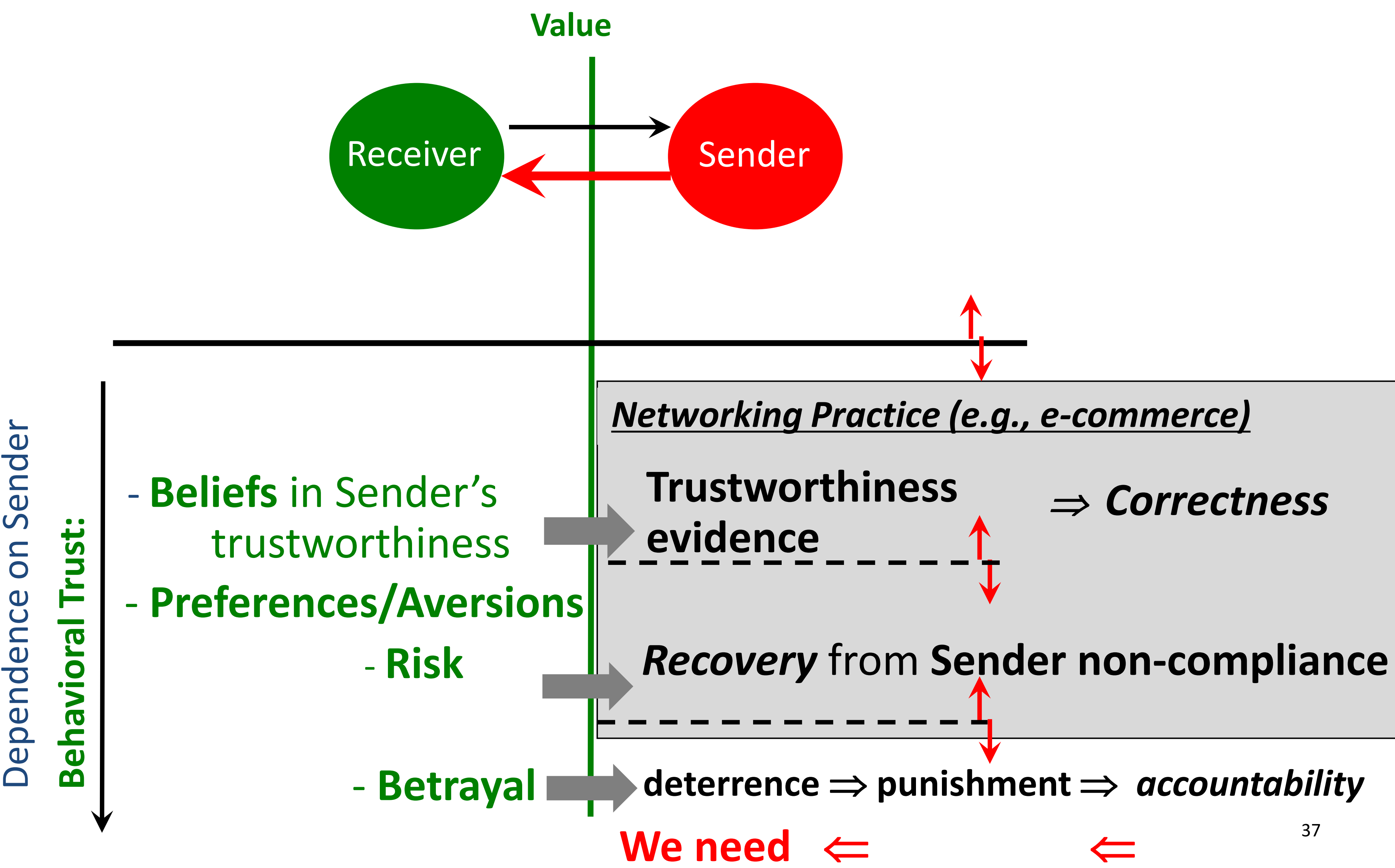
Dependence on Sender

Behavioral Trust:

- **Beliefs** in Sender's trustworthiness
- **Preferences/Aversions**
  - **Risk**
  - **Betrayal**

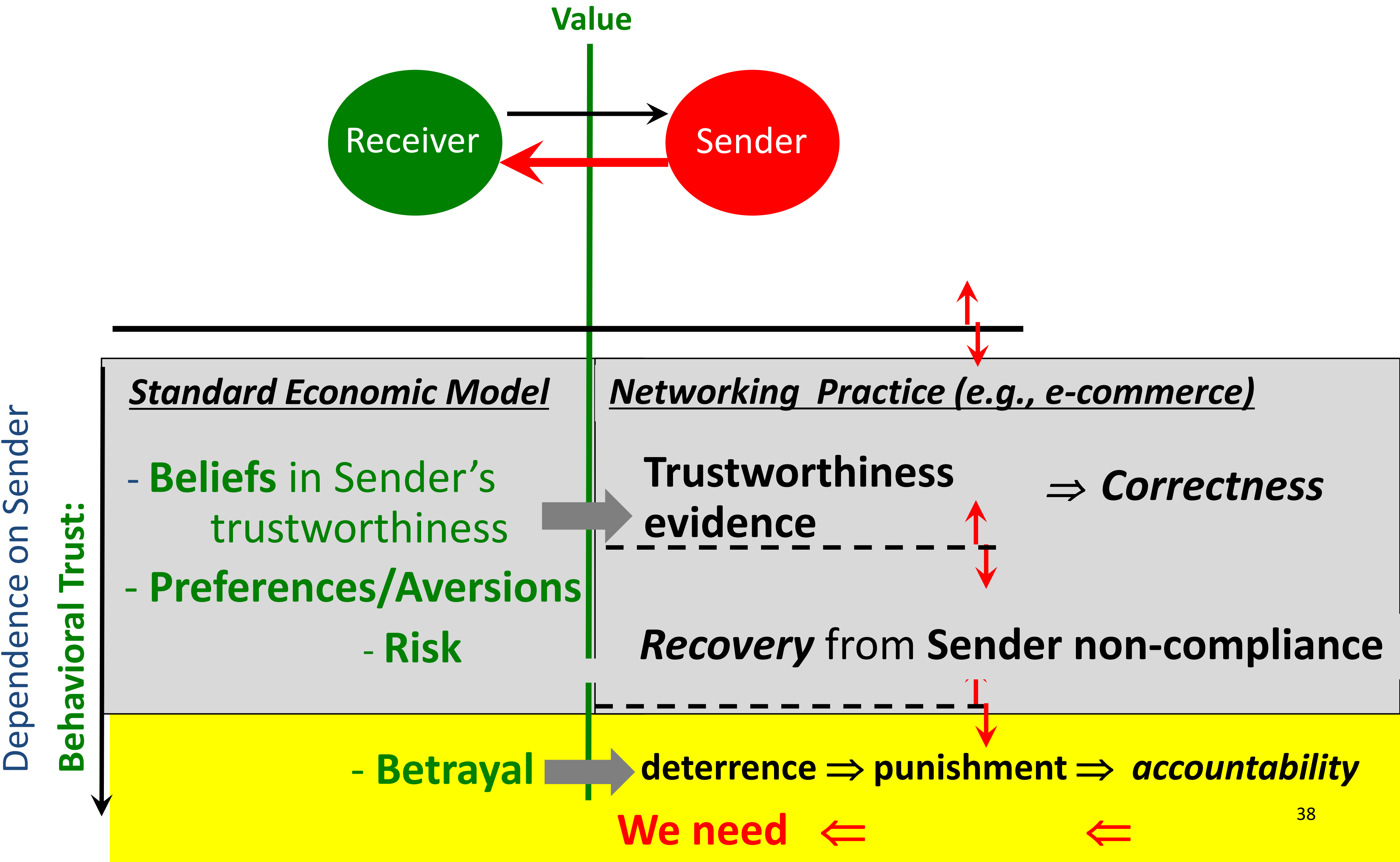
How can *all* these Primitives be Supported in Networks of Humans and Computers?

Relationship to Computational Trust Primitives



Need Primitives from Both Economics and Computing

Plus New Ones



Towards a (Richer) Theory of Trust:
New Approach for New Security Research

Past: Most security researchers have been merchants of fear.
We're good at it!

Future: Security infrastructures that promote *new* trust relations
(and cooperation)

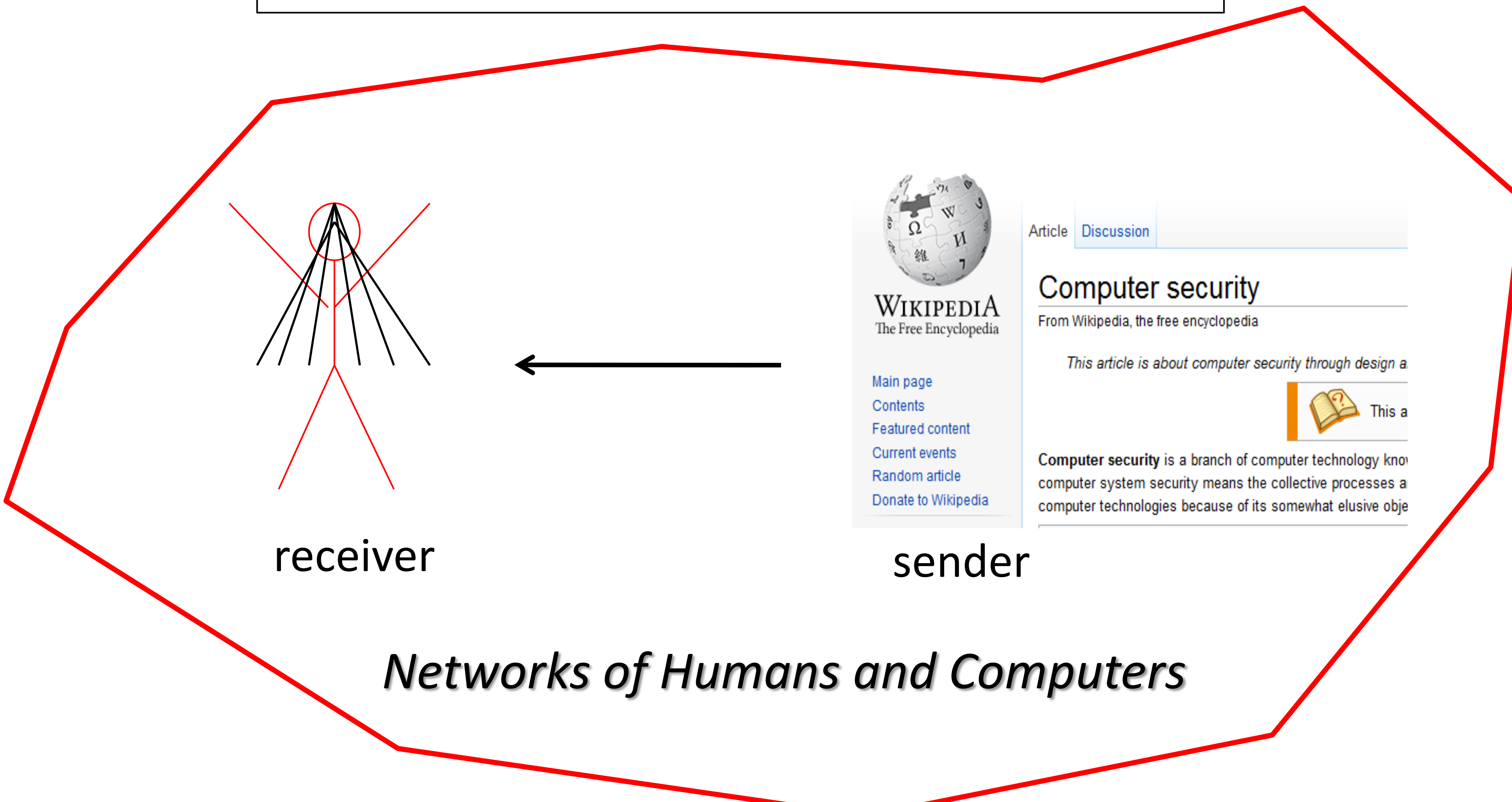
- Safety analogy:
  - air breaks in railcars (1896), automated railways signals and stops (1882)
    $\Rightarrow$ safe increase in train speeds, railroad commerce, economic opportunities

Goal: Seek security mechanisms that create new value, not just prevent losses

First Step: Behavioral Trust $\Rightarrow$ <u>closure for a class of trust primitives</u> for sender-receiver protocols

Motivation

How can I (a human) trust the information I read over the Internet?



Thank you!

References

- Akerlof, R.: Punishment, Compliance, and Anger in Equilibrium. Job Market Paper, MIT Sloan School, November 18 (2010) http://mit.academia.edu/RobertAkerlof/Papers/163148/Punishment_Compliance_and_Anger_in_Equilibrium_JOB_MARKET_PAPER_
- Biba, K. J. "Integrity Considerations for Secure Computer Systems", MTR-3153, The Mitre Corporation, April 1977.
- Clark, David D.; and Wilson, David R.; *A Comparison of Commercial and Military Computer Security Policies*; in *Proceedings of the 1987 IEEE Symposium on Research in Security and Privacy (SP'87), May 1987, Oakland, CA*; IEEE Press, pp. 184–193
- Fehr, E., Fischbacher, U., Kosfeld, M.: Neuroeconomic Foundations of Trust and Social Preferences. Forschungsinstitut zur Zukunft der Arbeit, IZA (Institute for the Study of Labor), Bonn, Germany (2005)
- Fehr, E.: The Economics and Biology of Trust. Journal of the European Economics Association (2009)
- de Quervain, D., Fishbacher, U., Treyer, V., Schellhammer, M., Schnyder, U., Buck, A., Fehr, E.: The Neural Basis for Altruistic Punishment. Science, Aug. 27 (2004)
- R Gennaro, C Gentry, B Parno Non-interactive verifiable computing: Outsourcing computation to untrusted workers, *Advances in Cryptology--CRYPTO 2010*, Springer
- Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In *the 41st ACM Symposium on Theory of Computing (STOC)*, 2009
- V. Gligor and J.M. Wing " Towards a Theory of Trust in Networks of Humans and Computers," in *Proceedings of Nineteenth International Workshop on Security Protocols*, Cambridge, England, March 28-30, 2011, to appear. **Invited Paper**.
- Lampson, B.W., \Usable Security: How to Get It," in Comm. ACM, Nov (2009)
- Bryan Parno, Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers, May 2010, CMU Ph.D. Dissertation, ACM Dissertation Award.