Deniable and not self-harming trapdoors

Rump Session of the *Crypto 2014 conference*
(August 19 – Santa Barbara, USA)

Luís Brandão* and René Peralta+

* Ph.D. student at FCUL-DI and CMU-ECE. Support for this research was provided by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program under Grant SFRH/BD/33770/2009.
+ National Institute of Standards and Technology

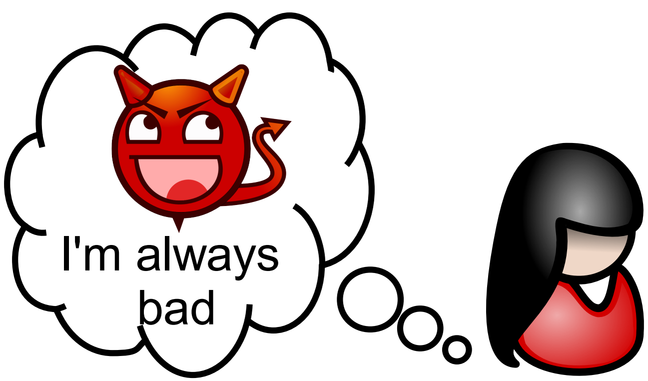
A simplistic adversarial model

(malicious Alice vs. honest-and-naive Bob)

A simplistic adversarial model

(malicious Alice vs. honest-and-naive Bob)

Assumption 1 (extra-knowledge): Alice knows more math than Bob



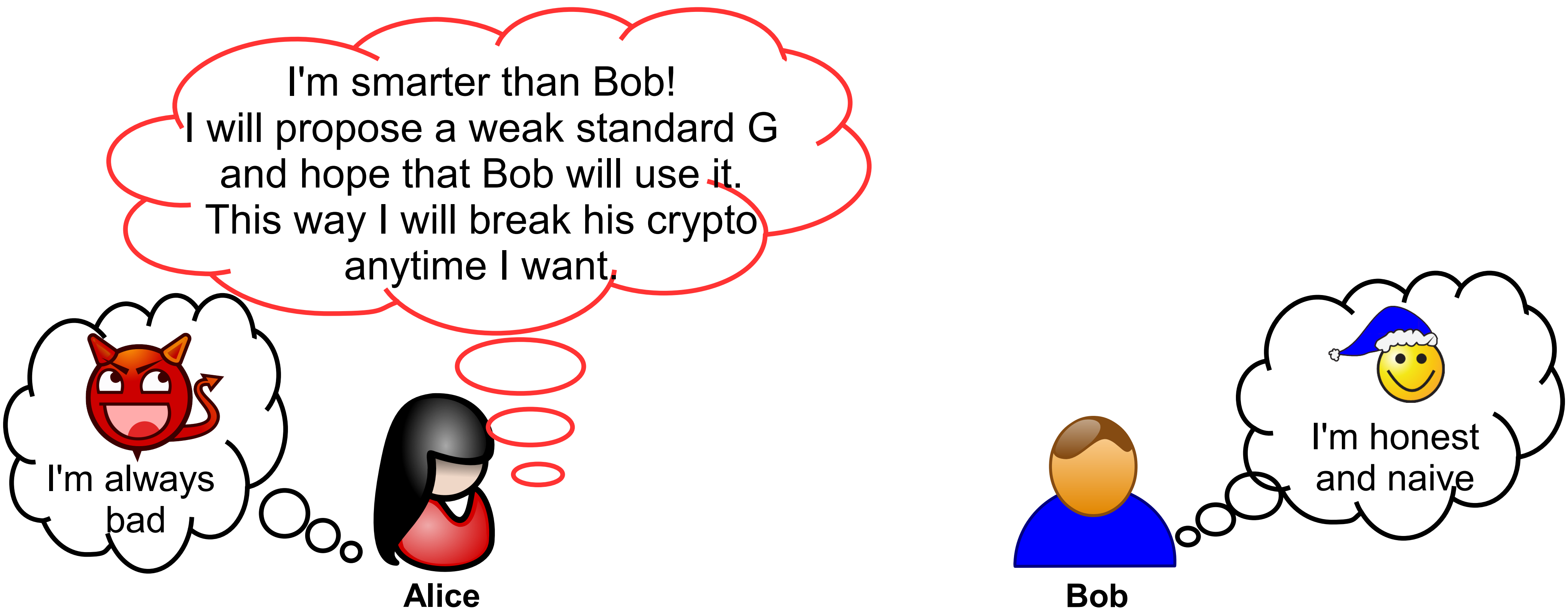
Alice

Bob

A simplistic adversarial model

(malicious Alice vs. honest-and-naive Bob)

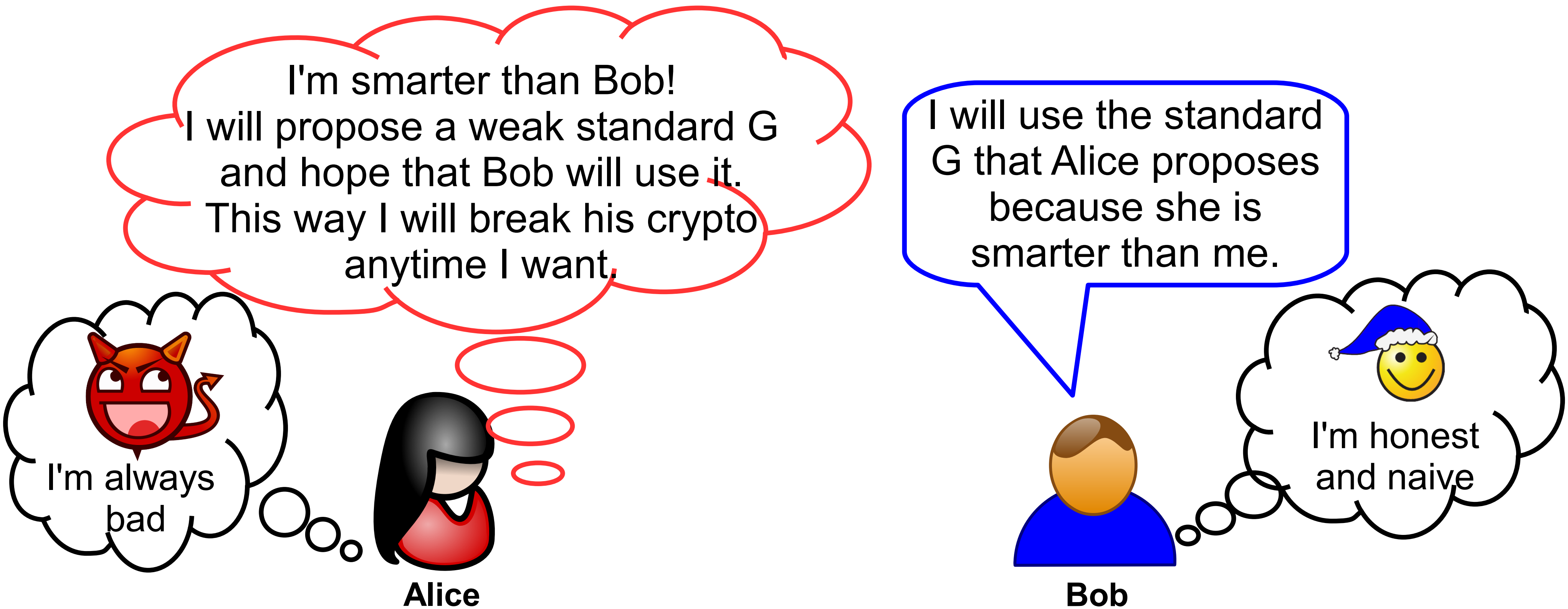
Assumption 1 (extra-knowledge): Alice knows more math than Bob



A simplistic adversarial model

(malicious Alice vs. honest-and-naive Bob)

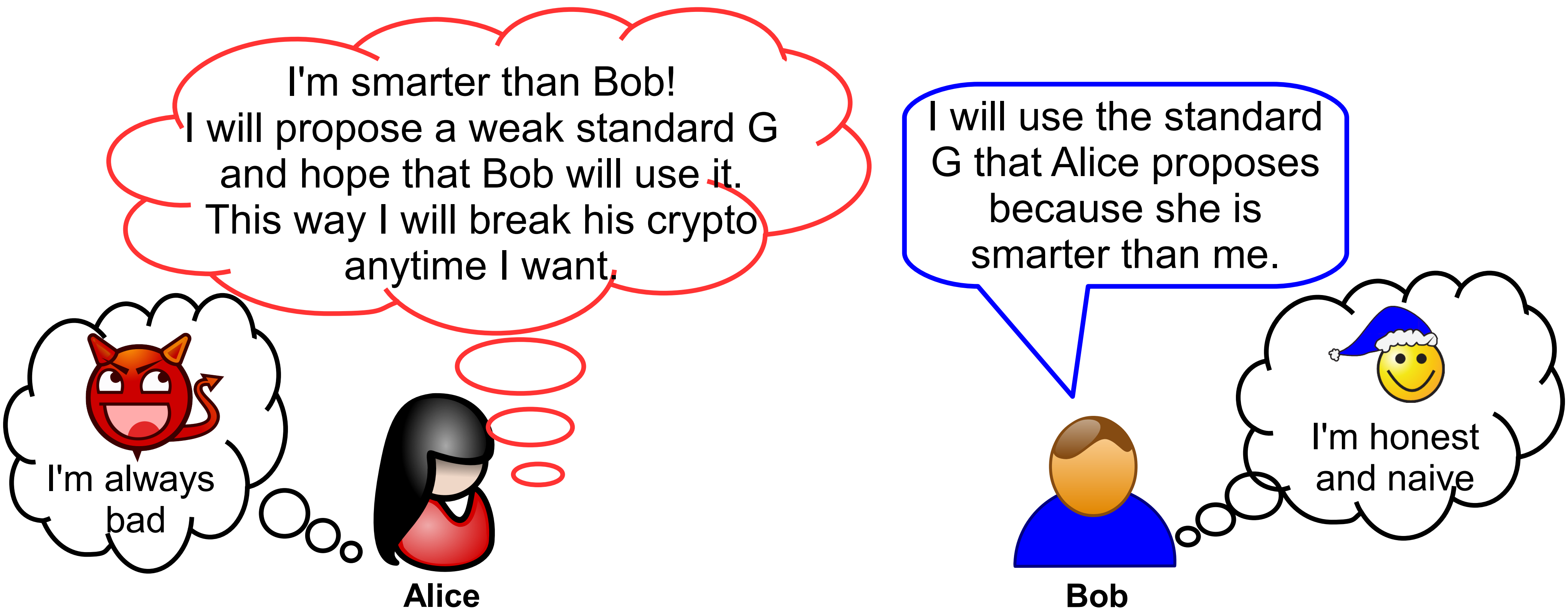
Assumption 1 (extra-knowledge): Alice knows more math than Bob



A simplistic adversarial model

(malicious Alice vs. honest-and-naive Bob)

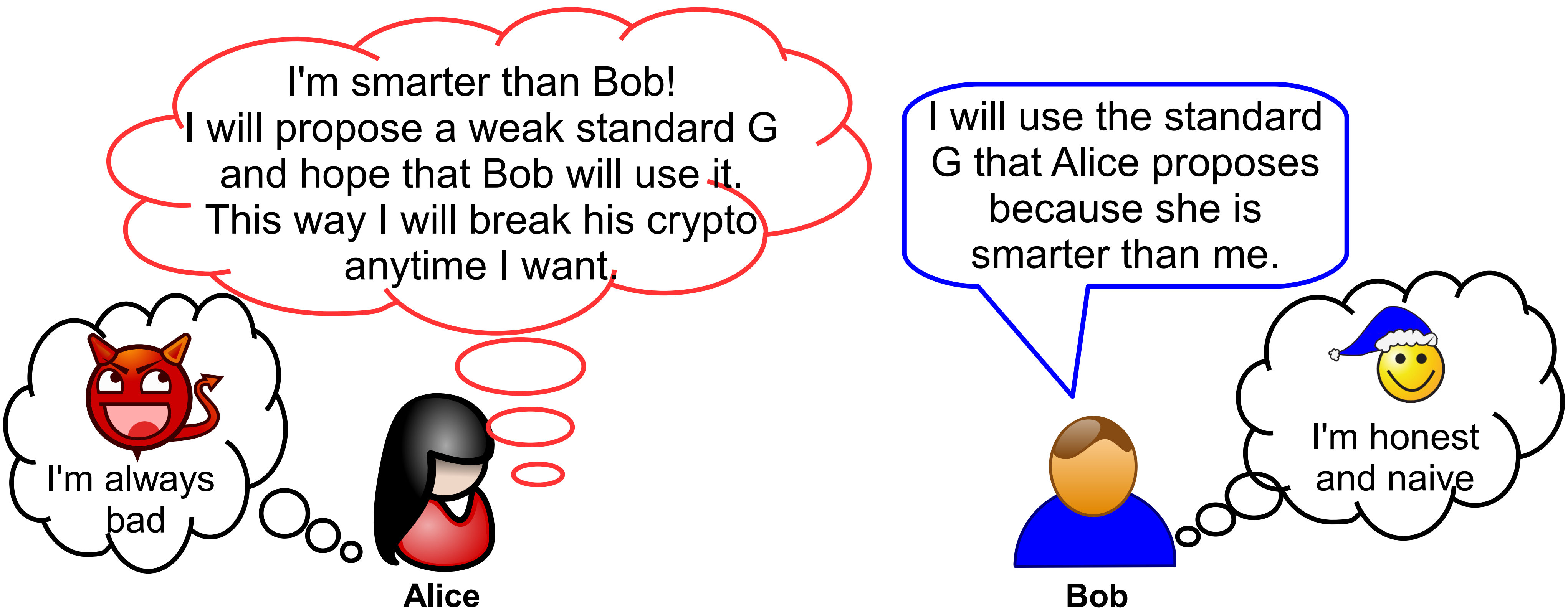
Assumption 1 (extra-knowledge): Alice knows more math than Bob



Example problem: Alice might be **the only one** knowing how to efficiently factor integers or compute discrete-logs in a particular type of groups.

A simplistic adversarial model
(malicious Alice vs. honest-and-naive Bob)

Assumption 1 (extra-knowledge): Alice knows more math than Bob



Example problem: Alice might be **the only one** knowing how to efficiently factor integers or compute discrete-logs in a particular type of groups.

But would Alice propose a knowingly-weak standard?

Another adversarial model

Semi-malicious Alice

Another adversarial model

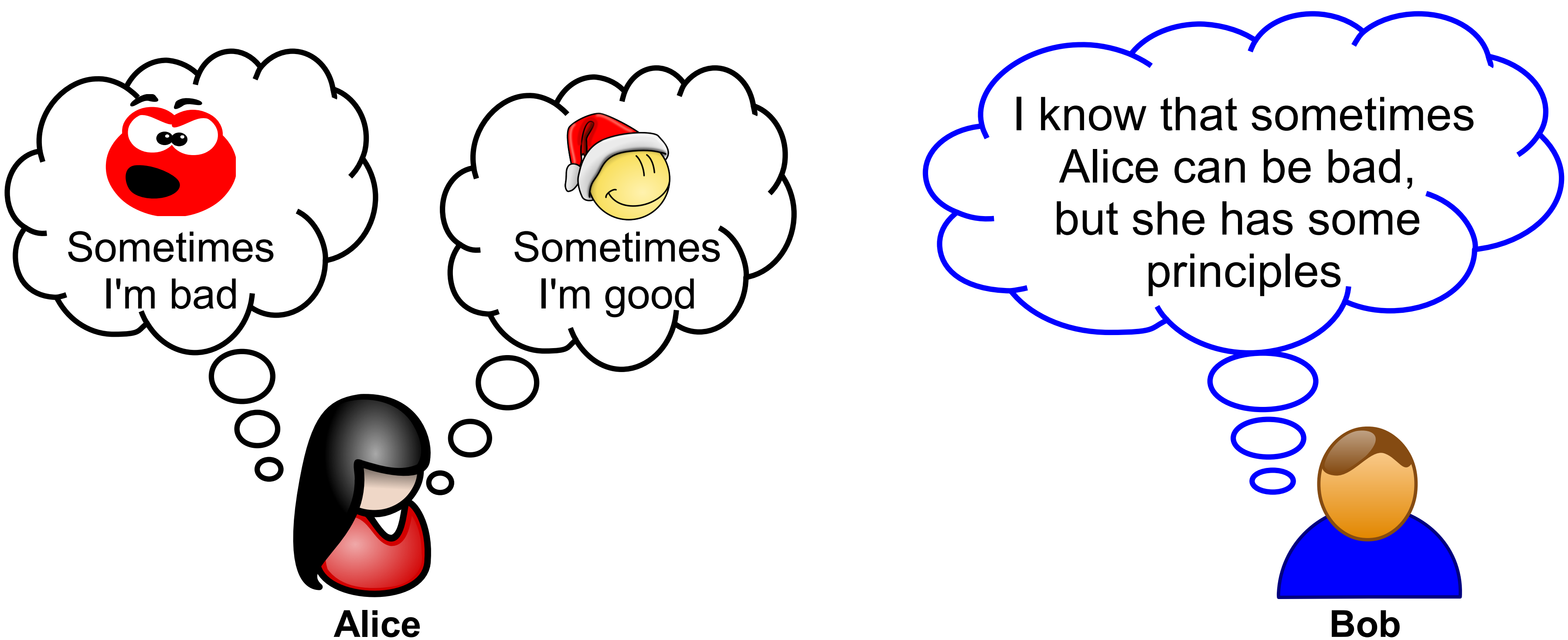
Semi-malicious Alice

i.e., malicious-but-with-principles (and very-curious)

Another adversarial model

Semi-malicious Alice

i.e., malicious-but-with-principles (and very-curious)

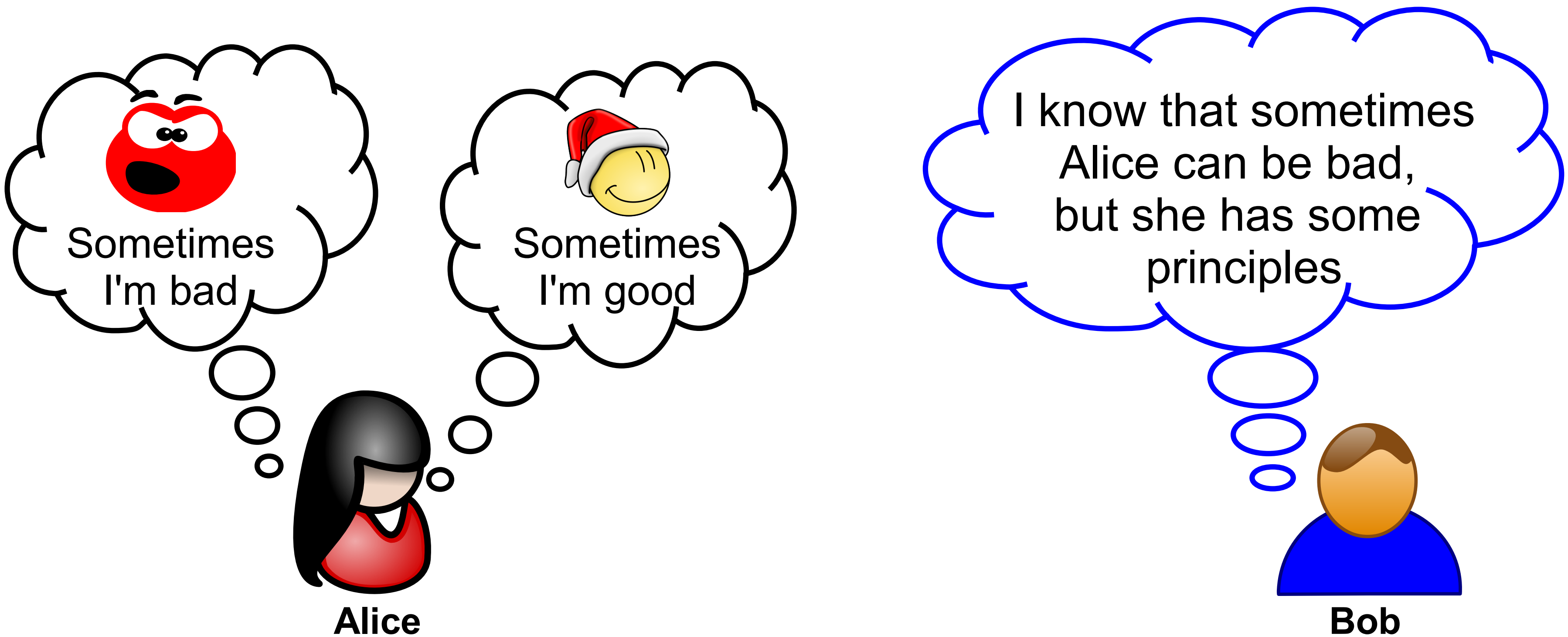


Another adversarial model

Semi-malicious Alice

i.e., malicious-but-with-principles (and very-curious)

Assumption 2 (team protection): Alice will not intentionally harm someone in her own team, but she still wants to break Bob's crypto.

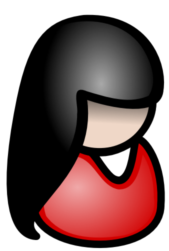


Alice selects good standard for her team

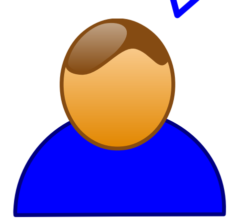
Alice selects good standard for her team

I'm the protector of my team (team-A). World, we can use G as a good standard!

We (team-B) will use standard G because we trust that Alice would not propose a standard that could harm her own team.



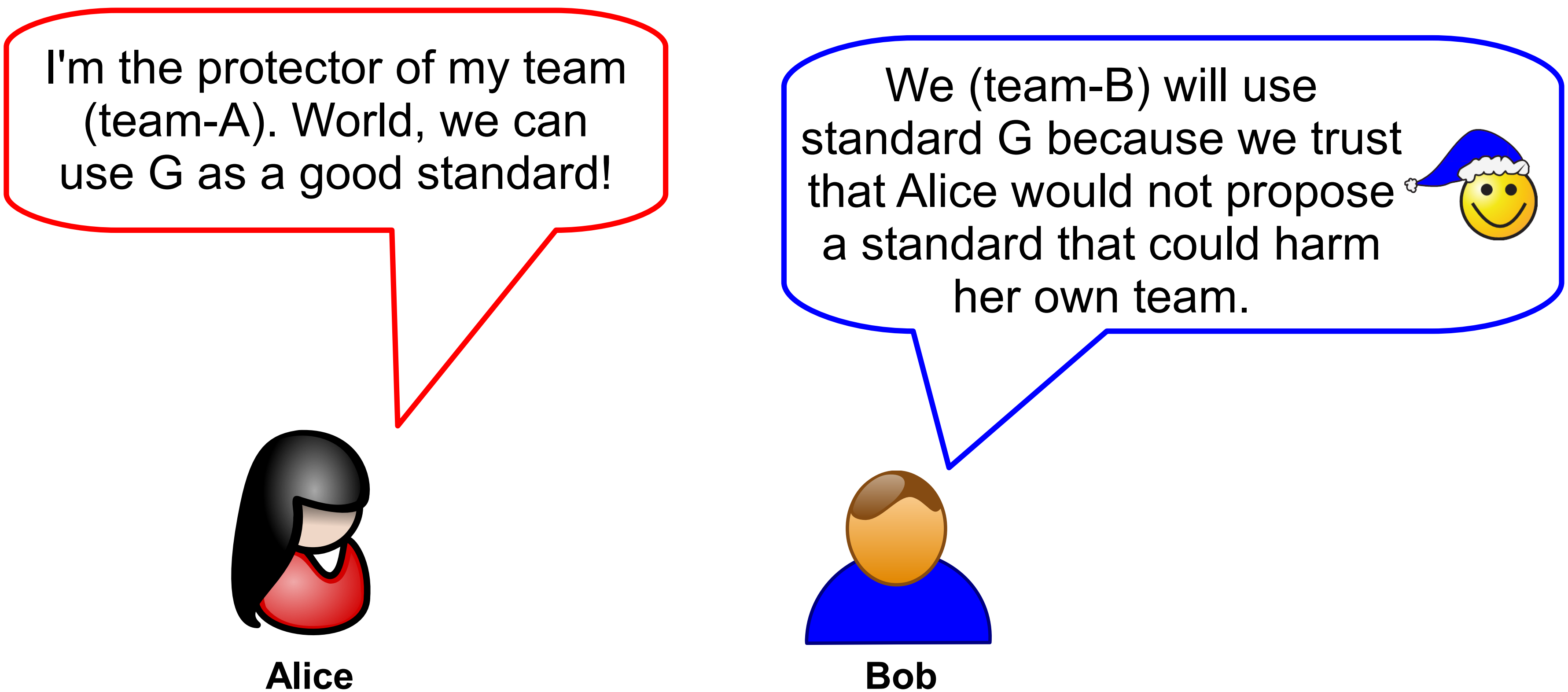
Alice



Bob

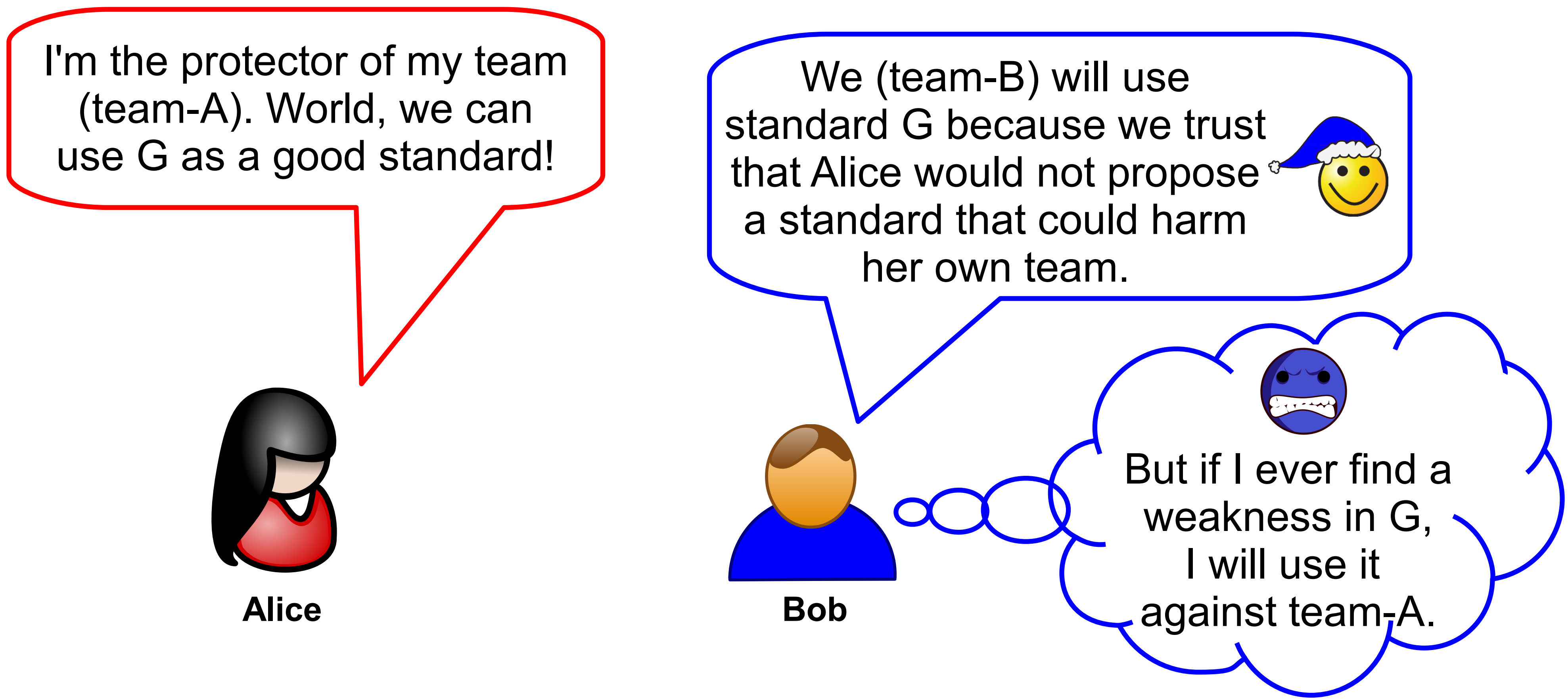
Alice selects good standard for her team

Assumption 3 (progressive-knowledge): The math that Alice knows now, Bob will eventually also learn in the future.



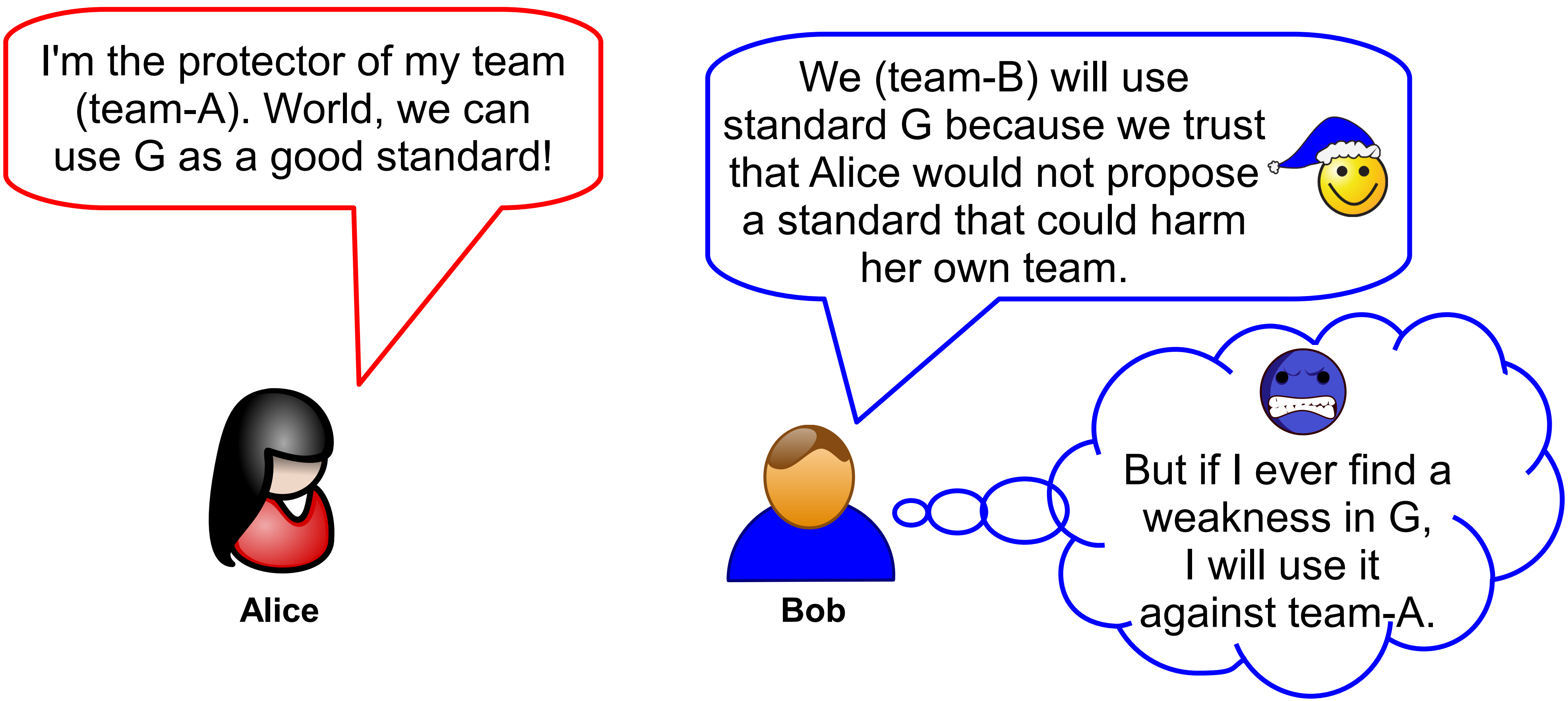
Alice selects good standard for her team

Assumption 3 (progressive-knowledge): The math that Alice knows now, Bob will eventually also learn in the future.



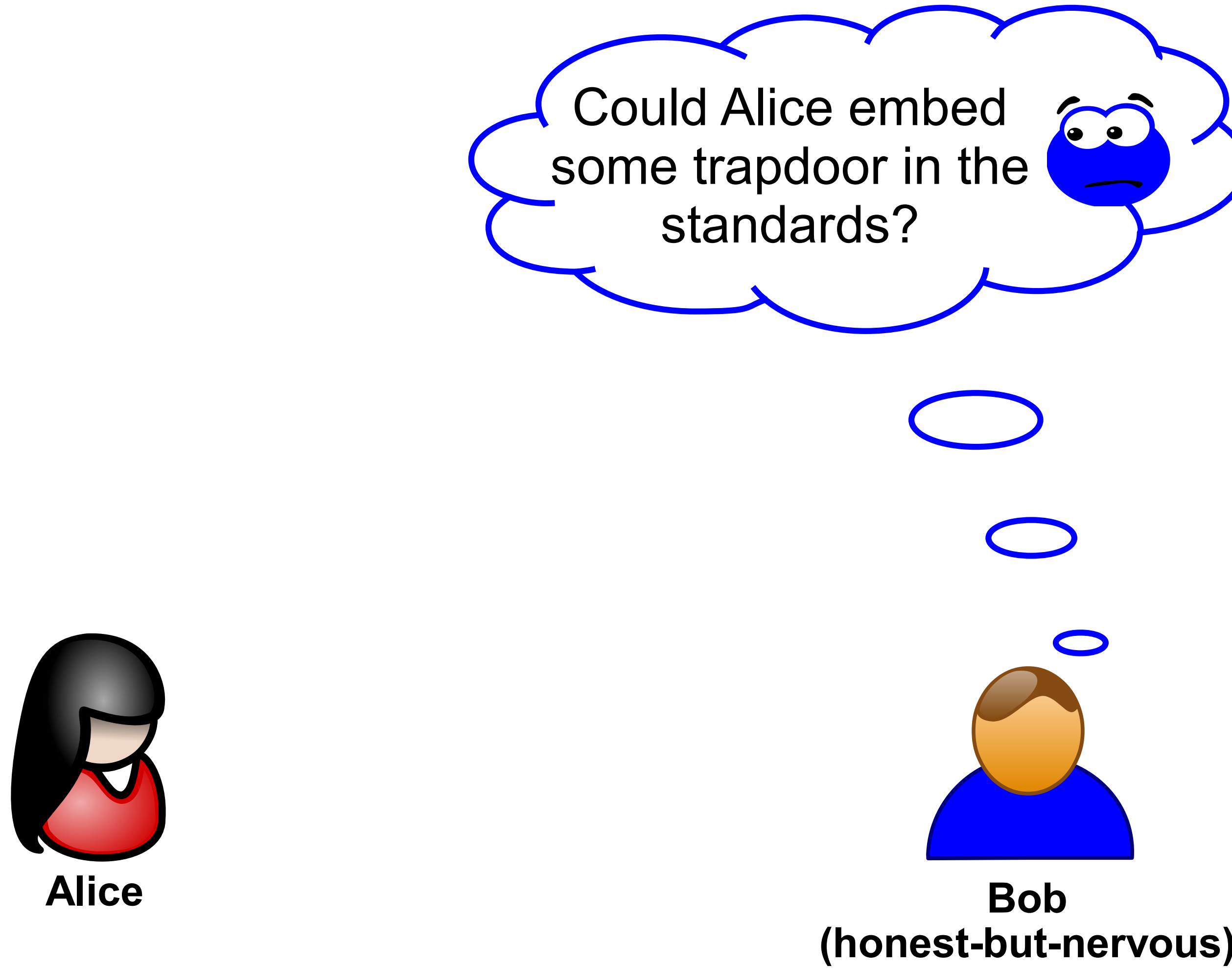
Alice selects good standard for her team

Assumption 3 (progressive-knowledge): The math that Alice knows now, Bob will eventually also learn in the future.



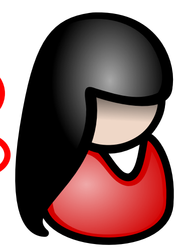
Under the assumptions, is the *standard* G good for Bob?

What about trapdoors?



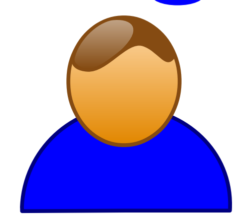
What about trapdoors?

Standard G is strong (no known weakness), but I know a trapdoor t to invert G. I can prove that finding the trapdoor is hard (or otherwise G is insecure beyond my knowledge)



Alice

Could Alice embed some trapdoor in the standards?

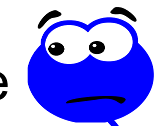


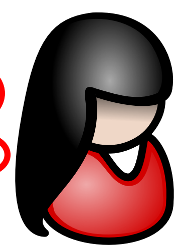
Bob
(honest-but-nervous)

What about trapdoors?

Standard G is strong (no known weakness), but I know a trapdoor t to invert G. I can prove that finding the trapdoor is hard (or otherwise G is insecure beyond my knowledge)

Could Alice embed some trapdoor in the standards?

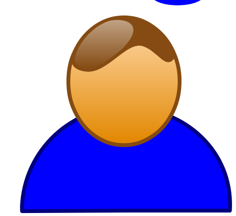




Alice

- $(G, t) \xleftarrow{\$} \mathrm{GEN}_{trap}(1^k)$
- $\mathrm{INV}(G,t) = G^{-1}$

(Efficient algorithms GEN_{trap} and INV may be known to Alice but unknown to Bob at this time)

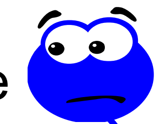


Bob (honest-but-nervous)

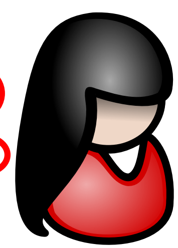
What about trapdoors?

Standard *G* is strong (no known weakness), but I know a trapdoor *t* to invert *G*. I can prove that finding the trapdoor is hard (or otherwise *G* is insecure beyond my knowledge)

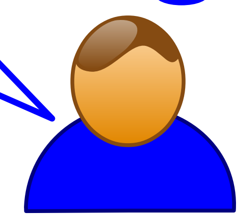
Could Alice embed some trapdoor in the standards?



Alice, please show me how you selected the (random) G.



Alice



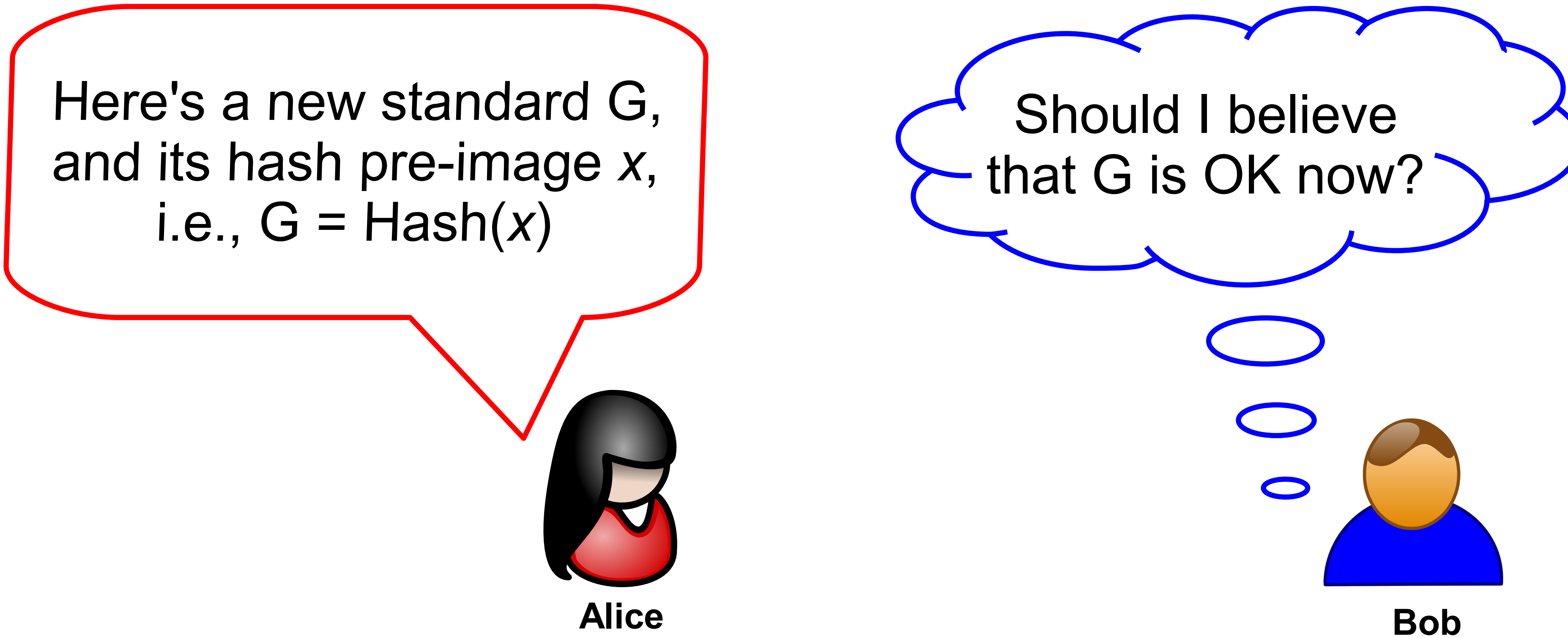
Bob
(honest-but-nervous)

- $(G, t) \xleftarrow{\$} \mathrm{GEN}_{\mathrm{trap}}(1^k)$
- $\mathrm{INV}(G,t) = G^{-1}$

(Efficient algorithms $\mathrm{GEN}_{\mathrm{trap}}$ and INV may be known to Alice but unknown to Bob at this time)

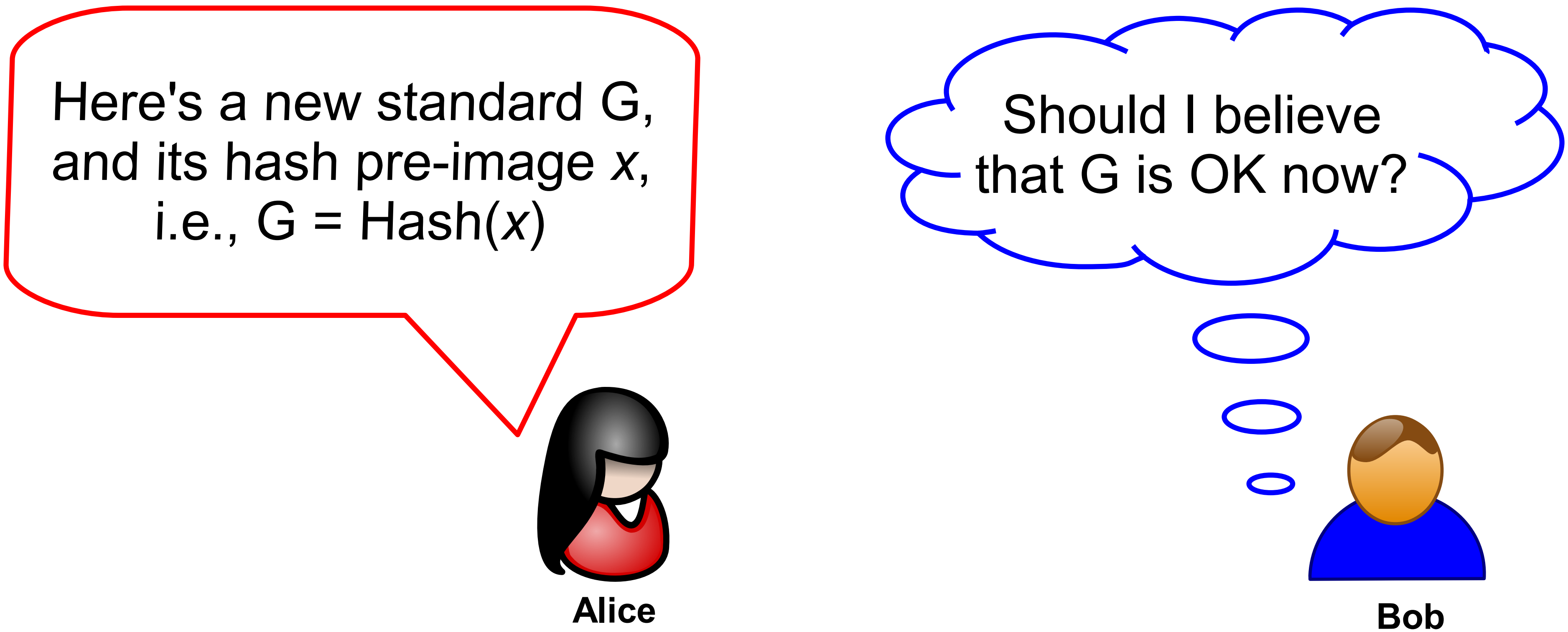
Standards with known hash pre-image

Alice tries to solve Bob's suspicion



Standards with known hash pre-image

Alice tries to solve Bob's suspicion



This prevents some trapdoor-embeddings, but maybe not all!

Hypothetical trapdoor embedding (for 128-bits security)

(Note: Alice needs to give pre-image of G, and G cannot be knowingly-weak)

Hypothetical trapdoor embedding (for 128-bits security)

(Note: Alice needs to give pre-image of G, and G cannot be knowingly-weak)

Hypothetically, Alice may know efficient algorithms (FG,FT, INV):

- (FG(G) == FT(t)) $\Rightarrow$ INV(G,t)=G^{-1}
- ($\forall$G) Prob[FG(G)==FG(t)] = 2^{-128} (for t selected after knowing G)

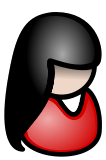
Hypothetical trapdoor embedding (for 128-bits security)

(Note: Alice needs to give pre-image of G, and G cannot be knowingly-weak)

Hypothetically, Alice may know efficient algorithms (FG,FT, INV):

- $(\mathrm{FG}(\mathrm{G}) == \mathrm{FT}(t)) \Rightarrow \mathrm{INV}(\mathrm{G},t)=\mathrm{G}^{-1}$
- $(\forall \mathrm{G})$ Prob$[\mathrm{FG}(\mathrm{G})==\mathrm{FG}(t)] = 2^{-128}$ (for t selected after knowing G)

Using $\tilde{\mathrm{O}}(2^{64})$ ops, Alice can generate standard with trapdoor:



Alice

1) For i=1,...,$\tilde{\mathrm{O}}(2^{64})$: $a_i \xleftarrow{\$} \{0,1\}^{128}$ (pre-image) , $\mathrm{G}_i = \mathrm{Hash}(a_i)$
2) For i=1,...,$\tilde{\mathrm{O}}(2^{64})$: $t_i \xleftarrow{\$} \{0,1\}^{128}$ (tentative trapdoor)
3) Using $\tilde{\mathrm{O}}(2^{64})$ ops, find (i,j) : $\mathrm{FG}(\mathrm{G}_i)==\mathrm{FT}(t_j)$ – then let $\mathrm{G}\equiv\mathrm{G}_i$ and $t\equiv t_j$.
4) Then, INV efficiently computes G^{-1}, e.g., $x = \mathrm{INV}(G,t,g,g^x)$

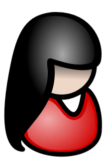
Hypothetical trapdoor embedding (for 128-bits security)

(Note: Alice needs to give pre-image of G, and G cannot be knowingly-weak)

Hypothetically, Alice may know efficient algorithms (FG,FT, INV):

- $(FG(G) == FT(t)) \Rightarrow INV(G,t)=G^{-1}$
- $(\forall G)$ Prob[FG(G)==FG(t)] $= 2^{-128}$ (for t selected after knowing G)

Using $\tilde{O}(2^{64})$ ops, Alice can generate standard with trapdoor:



Alice

1) For i=1,...,$\tilde{O}(2^{64})$: $a_i \xleftarrow{\$} \{0,1\}^{128}$ (pre-image) , $G_i = \text{Hash}(a_i)$
2) For i=1,...,$\tilde{O}(2^{64})$: $t_i \xleftarrow{\$} \{0,1\}^{128}$ (tentative trapdoor)
3) Using $\tilde{O}(2^{64})$ ops, find (i,j) : $FG(G_i)==FT(t_j)$ – then let $G \equiv G_i$ and $t \equiv t_j$.
4) Then, INV efficiently computes G^{-1}, e.g., $x = INV(G,t,g,g^x)$

The standard is "strong" (for team-A) and the trapdoor is "deniable"

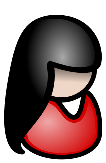
Hypothetical trapdoor embedding (for 128-bits security)

(Note: Alice needs to give pre-image of G, and G cannot be knowingly-weak)

Hypothetically, Alice may know efficient algorithms (FG,FT, INV):

- $(\mathrm{FG}(G) == \mathrm{FT}(t)) \Rightarrow \mathrm{INV}(G,t)=G^{-1}$
- $(\forall G)$ $\mathrm{Prob}[\mathrm{FG}(G)==\mathrm{FG}(t)] = 2^{-128}$ (for t selected after knowing G)

Using $\tilde{O}(2^{64})$ ops, Alice can generate standard with trapdoor:



Alice

1) For i=1,...,$\tilde{O}(2^{64})$: $a_i \xleftarrow{\$} \{0,1\}^{128}$ (pre-image) , $G_i = \mathrm{Hash}(a_i)$
2) For i=1,...,$\tilde{O}(2^{64})$: $t_i \xleftarrow{\$} \{0,1\}^{128}$ (tentative trapdoor)
3) Using $\tilde{O}(2^{64})$ ops, find (i,j) : $\mathrm{FG}(G_i)==\mathrm{FT}(t_j)$ – then let $G \equiv G_i$ and $t \equiv t_j$.
4) Then, INV efficiently computes G^{-1}, e.g., $x = \mathrm{INV}(G,t,g,g^x)$

- **Strong:** Even when Bob catches up on the math of Alice, he can still not find the trapdoor (it would require 2^{128} ops).
- **Deniable:** Alice can pretend that she did not know (FG,FT,INV) at the time of creating G (which is indeed being uniformly selected).

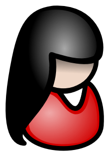
Conclusions

Even a “team-protector” might be able to embed a trapdoor in a deniable way and without harming her own team.

(cost ≈ square-root of cost of finding t after seeing G)

Further interesting considerations:

Conclusions

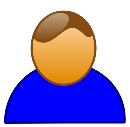


Alice

Even a “team-protector” might be able to embed a trapdoor in a deniable way and without harming her own team.

(cost ≈ square-root of cost of finding t after seeing G)

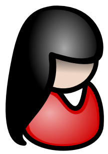
Further interesting considerations:



Bob

- **Defender: How to prevent deniable trapdoors?** (despite “extra-knowledge” by Alice, but within the “team-protection” and “progressive knowledge” assumptions)

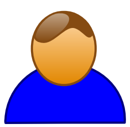
Conclusions


Alice

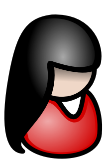
Even a “team-protector” might be able to embed a trapdoor in a deniable way and without harming her own team.

(cost ≈ square-root of cost of finding t after seeing G)

<u>Further interesting considerations:</u>

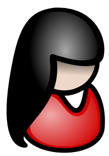

Bob

- <u>**Defender:**</u> **How to prevent deniable trapdoors?** (despite “extra-knowledge” by Alice, but within the “team-protection” and “progressive knowledge” assumptions)


Alice

- <u>**Attacker:**</u> **How to embed *deniable trapdoors more-efficiently*?** (despite having to show hash pre-images)

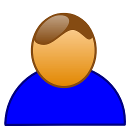
Conclusions


Alice

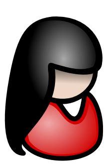
Even a "team-protector" might be able to embed a trapdoor in a deniable way and without harming her own team.

(cost ≈ square-root of cost of finding t after seeing G)

<u>Further interesting considerations:</u>


Bob

- **<u>Defender:</u> How to prevent deniable trapdoors?** (despite "extra-knowledge" by Alice, but within the "team-protection" and "progressive knowledge" assumptions)


Alice

- **<u>Attacker:</u> How to embed *deniable trapdoors more-efficiently*?** (despite having to show hash pre-images)

Thank you for your attention!