On Robust Key Agreement Based on Public Key Authentication

Speaker: Feng Hao

Thales E-Security, Cambridge, UK

Financial Cryptography'10

Outline

Provable security in a formal model

- Formal analysis of a cryptography protocol.
  1. Formal adversary model
  2. Formal security definitions
  3. Formal security proofs
- Nowadays, almost every protocol is “provably secure”.
- But, we need to interpret this carefully.

Debates between MQV and HMQV

- MQV is a widely standardized key agreement protocol.
- HMQV is modified from MQV with aim for provable security.
- Seen as a prime example of success of formal analysis.
- HMQV has formal proofs while MQV doesn't.
- So, HMQV must be more secure. No?
- In fact, a complicated issue ...

HMQV protocol

Alice $(\hat{A}, g^a)$		Bob $(\hat{B}, g^b)$
1. $x \in_R Z_q$	$\xrightarrow{X = g^x}$	Verify $X \neq 0$
2. Verify $Y \neq 0$	$\xleftarrow{Y = g^y}$	$y \in_R Z_q$
$d = \bar{H}(X, \hat{B}),\ e = \bar{H}(Y, \hat{A})$		
Alice computes: $\kappa = H((YB^e)^{x+da}) = H(g^{(x+da)(y+eb)})$		
Bob computes: $\kappa = H((XA^d)^{y+eb}) = H(g^{(x+da)(y+eb)})$		

- Changes from MQV
  1. Using hash functions
  2. Removing both the static and ephemeral public key validations

HMQV protocol - revised in IEEE P1363

- HMQV dispenses validating g^a and g^x other than not 0.
- But, a small subgroup attack reveals the private key.
- This shows a serious flaw in the HMQV security model.
- In submission to IEEE P1363, HMQV was revised.
- The revision adds validating $XA^d = g^x g^{ad}$.
- Unfortunately, the revised HMQV still subject to attacks.
- We will show two new attacks.

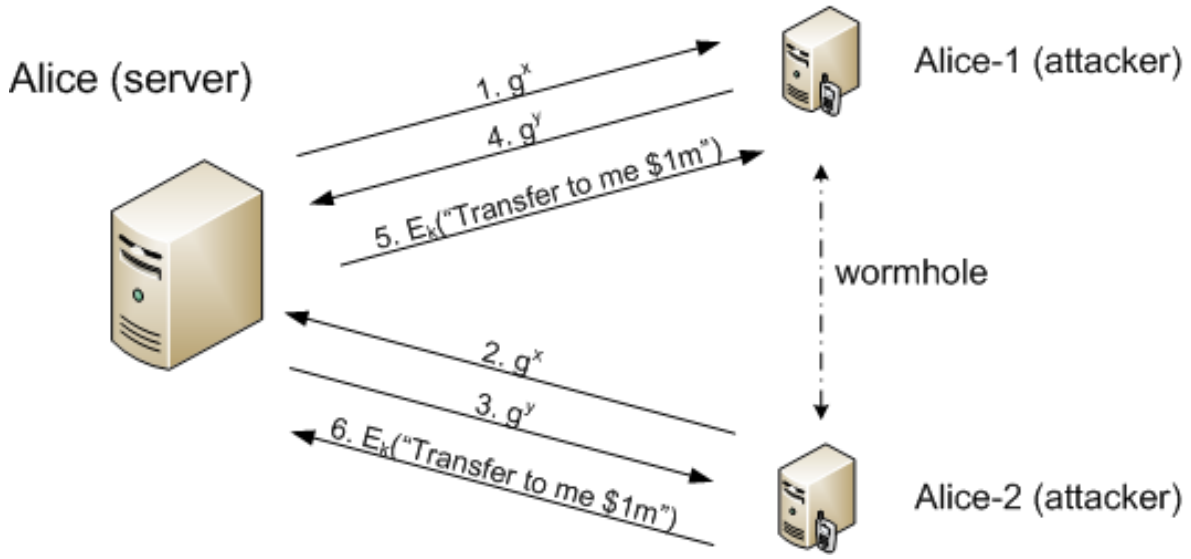
Invalid Public Key attack on HMQV

- Bob registers a small subgroup element s as the public key.
- CA checks s is not zero and certify it (HMQV specification).
- Bob does some precomputation (details in paper).
- Now he can successfully authenticate to Alice.
- But, Bob doesn't even have a private key!

Invalid Public Key attack - where goes wrong?

- A small subgroup element s is clearly an invalid public key.
- There does not even exist a private key.
- Anyone who knows s can pass authentication successfully.
- This shows HMQV doesn't fulfill the basic definition of authentication.
- The attack not applicable to MQV.

Wormhole Replay attack on HMQV



- Self-communication is formally proven "secure" in HMQV.
- A station and its mobile clients use the same certificate.
- Attacker creates two authenticated channels without the private key.

Wormhole Replay attack - where goes wrong?

- The HMQV model implicitly assumes: only one mobile client.
- But, in practice, there can be several mobile clients.
- This is a common deficiency in all current formal models.
- Applicable to NAXOS, KEA+, CMQV, MQV, and SIG-DH etc.

Which is the right security model?

- Many attacks are related to deficiencies in the model.
- In key agreement, several models: CK, eCK, HMQV-CK etc.
- But, which one is the "right" model?
- Argued for many years, still no consensus.
- Each model only defines a subset of attacker's abilities.
- Theoretical comparisons.
  - Protocols proven secure in CK may prove insecure in eCK.
  - Protocols proven secure in eCK may prove insecure in CK.
- Also, there are practical attacks on all models.

A different approach - YAK protocol

You change the problem if you can't solve it.

– David Wheeler

Basic design ideas in YAK

- Don't model the adversary.
- Assume an extreme adversary
  - The only powers he doesn't have are those that would allow him to trivially break any other protocol.
- Adopt prudent engineering principles, such as
  - The sixth robustness principle (Anderson-Needham, Crypto'95)
  - The explicitness principle
- Most importantly, keep it simple.

YAK protocol

- CA registration (PKI standard)
  - Alice and Bob register g^a and g^b with knowledge proofs for the private keys.
- YAK key agreement
  - Alice sends out g^x with a knowledge proof for x.
  - Bob sends out g^y with a knowledge proof for y.
  - Compute session key: $\kappa = H(g^{(x+a)(y+b)})$.

Security properties

1. **Private key security**: An attacker cannot learn any useful information about the user's static private key even if he is able to learn all session specific secrets in any session.
2. **Full forward secrecy**: Session keys that were securely established in the past uncorrupted sessions will remain secure in the future even when both users' static private keys are disclosed.
3. **Session key security**: An attacker cannot compute the session key if he impersonates a user but has no access to the user's private key.

- The revised HMQV (IEEE P1363) doesn't fulfill the third.

Performance of YAK

- Some people think zero-knowledge proof too expensive.
- But, the cost depends on how effective is the integration.
- We use Schnorr signature as an example.
- Effectively, YAK requires 4 exponentiations.
- In comparison, MQV/HMQV need 3.5.

Conclusion

- Provable security is a tool, not the answer.
- Showed two new attacks on HMQV.
  - First attack invalidates the basic authentication in HMQV.
  - Second attack applies to many other "provably secure".
- Presented a new key agreement protocol: YAK.
  - Robust against an extreme adversary.
  - Comparable computational efficiency to MQV/HMQV.
  - So far, the simplest among all related protocols.